

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.896

До захисту допущено
В. о. завідувача кафедри ММСА

О.Л.Тимошук

«___» _____ 2018 р.

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 122 Комп'ютерні науки
на тему: «Застосування методів штучного інтелекту в задачі розпізнаванні
об'єктів на аерофотознімках»

Виконав:

студент II курсу, групи КА-73 мп
Величко Сергій Вікторович

Керівник: Професор кафедри ММСА
д.т.н., професор, Зайченко Ю.П.

Рецензент: Старший науковий співробітник
кафедри програмного забезпечення комп'ютерних систем
КПІ ім. Ігоря Сікорського
д.т.н., с.н.с, Вішталъ Д.М.

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів
без відповідних посилань
Студент _____

Київ
2018

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)

Спеціальність (спеціалізація) — 122 «Комп'ютерні науки» («Системи штучного інтелекту»)

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри ММСА

О. Л. Тимощук

«___» _____ 2018 р.

ЗАВДАННЯ

на магістерську дисертацію студенту Величку Сергію Вікторовичу

1. Тема дисертації: «Застосування методів штучного інтелекту в задачі розпізнаванні об'єктів на аерофотознімках», науковий керівник дисертації Зайченко Юрій Петрович, доктор технічних наук, професор кафедри ММСА, затверджені наказом по університету від «07» листопада 2018 р. № 4121-с

2. Термін подання студентом дисертації: _____

3. Об'єкт дослідження: обробка супутникових знімків

4. Предмет дослідження: алгоритми автоматичної сегментації та розпізнання об'єктів на знімках

5. Перелік завдань, які потрібно розробити:

1) провести аналіз методів штучного інтелекту для вирішення прикладних задач;

2) провести аналіз існуючих методів для вирішення задач сегментації зображення;

3) вибрати навчальний набір даних;

4) синтезувати архітектуру нейронної мережі;

5) провести навчання отриманої нейронної мережі

6. Дата видачі завдання: _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Написання вступу магістерської дисертації	07.09.2018—13.09.2018	
2.	Порівняльний аналіз підходів по вирішенню поставленої задачі, вибір архітектури моделі	14.09.2018—23.09.2018	
3.	Підготовка навчальної вибірки	24.09.2018—30.09.2018	
4.	Аналіз архітектур нейронних мереж для задачі сегментації зображень	01.10.2018—08.10.2018	
5.	Навчання нейронної мережі	09.10.2018—17.10.2018	
6.	Підготовка матеріалів першого розділу магістерської дисертації	18.10.2018—26.10.2018	
7.	Підготовка матеріалів другого розділу магістерської дисертації	27.10.2018—04.11.2018	
8.	Підготовка матеріалів третього розділу магістерської дисертації	05.11.2018—14.11.2018	
9.	Підготовка матеріалів розділу стартап-проекту магістерської дисертації	15.11.2018—22.11.2018	
10.	Написання висновку магістерської дисертації	23.11.2018—26.11.2018	

Студент

С.В. Величко

Науковий керівник дисертації

Ю.П. Зайченко

РЕФЕРАТ

Магістерська дисертація: 110 с., 48 рис., 15 табл., 26 джерел літератури.

Актуальність теми: сучасні методи аерофотозйомки, і машинного навчання здатні раз і назавжди вирішити проблему застарілих карт, завдяки точним знімкам і коректному розпізнаванню об'єктів на них з'являється можливість автоматизувати і значно прискорити процес складання карт, в тому числі екстрених і військових.

Мета і задачі дослідження: метою даної роботи є дослідження технології сегментації супутникових знімків.

Досягнуті результати: в ході виконання даної роботи було отримано неоціненний досвід роботи з засобами машинного навчання

Методи досліджень: для вирішення проблем в даній роботі використовувалися методи аналізу, синтезу, системного аналізу, порівняння, логічного узагальнення.

Наукова новизна: наукова новизна даної роботи полягає в пошуку нових підходів до навчання згорткової нейронної мережі для досягнення найкращих результатів.

Практичне значення отриманих результатів: отримані результати можуть бути застосовані в низці областей так чи інакше пов'язаних з картографією: екстрена картографія, контроль зелених насаджень, моніторинг стану полів і т.д.

АЕРОФОТОЗЙОМКА, КАРТОГРАФІЯ, ПАНОРАМНИЙ ЗНІМОК, СЕГМЕНТАЦІЯ, НЕЙРОННА МЕРЕЖА, U-NET, PYTHON, TENSORFLOW.

ABSTRACT

Master thesis takes 110 pages, contains 48 illustrations and 15 tables, was used 26 sources during preparations.

Topicality of subject: cartography – that where is combination of machine learning and cartography coming handy. Machine learning algorithms allows us to recognize any changes on terrain photo and put it to the map immediately. So, now we have a chance to solve such problem like irrelevant maps for good.

Aim and goals of research: the goal of these research is to find an optimal way to educate neural network for object recognition.

Achievements of research: the research provided author with valuable experience and concluded with optimal machine learning model trained.

Methods of research: problems, which are represented in these research were solved by using following methods: analyses, synthesis, survey, logical comparison, logical generalization.

Scientific innovation: scientific innovation of current research lies in finding new approaches to convolutional neural network education.

Application of results: results of current research might be applied in various aspects of human activity which are connected to cartography. Its might commit multiple monitoring tasks including, emergency maps generation, forest resource observation, crops observation etc.

AERIAL PHOTOGRAPHY, CARTOGRAPHY, PANORAMIC PHOTO, IMAGE SEGMENTATION, NEURAL NETWORK, U-NET, PYTHON, TENSORFLOW.

ЗМІСТ

РЕФЕРАТ	4
ABSTRACT.....	5
ВСТУП.....	8
1 ВВЕДЕННЯ У ПРЕДМЕТНУ ОБЛАСТЬ.....	9
1.1 Актуальність	9
1.2 Системи зондування земної поверхні.....	21
1.3 Задача сегментації зображення.....	28
1.4 Нейронні мережі	43
1.5 Згорткові нейронні мережі	45
Висновки до першого розділу	50
2 ОГЛЯД ДОСЛІДЖЕННЯ.....	52
2.1 Вхідні дані	52
2.2 Архітектура нейронної мережі U-net	58
2.2.1 ReLU	60
2.2.2 Згортковий шар	61
2.2.3 Субдіскретизуючий шар	63
2.3 Метрики оцінювання	64
2.4 Навчання нейронної мережі U-net.....	65
2.5 Огляд інструментів	68
2.6 Огляд отриманих результатів	69
2.7 Підсумки проведеного дослідження	75

Висновки до другого розділу	77
3 СТАРТАП	79
3.1 Передмова.....	79
3.2 Пошук та аналіз ідей	80
3.3 Аналіз ринку.....	85
3.4 Ринкова стратегія проекту	94
3.5 Маркетингова концепція	97
Висновки до третього розділу	98
ВИСНОВКИ ПО РОБОТІ.....	100
ПЕРЕЛІК ПОСИЛАНЬ	102
ДОДАТОК А – ЛІСТИНГ	105

ВСТУП

З самих витоків людської цивілізації існувала необхідність орієнтуватися на місцевості, яка в міру встановлення зв'язків між громадами і розширення поля діяльності лише зростала. Таким чином виникало рішення, у вигляді картографії - люди наносили помітні об'єкти, нерівності місцевості, дороги і селища на карти, що істотно знижувало вірогідність заблукати.

Автоматичне розпізнавання об'єктів на знімках місцевості дозволить як складати максимально точні карти місцевості в короткі терміни, так і проводити моніторинг різного роду людської діяльності: визначати стан посівів на полях, аналізувати завантаженість автомобільних доріг, знаходити різного роду незаконні споруди і так далі.

Метою даної роботи є створення моделі та алгоритму сегментації для розпізнавання об'єктів на супутникових знімках та пошук можливих напрямів її впровадження. В ході роботи було поставлено та вирішено наступні задачі:

- 1) дослідження існуючих підходів для вирішення поставленої задачі;
- 2) застосування методів та засобів штучного інтелекту у задачі сегментації об'єктів на аерофотознімках;
- 3) проектування та розробка програмного комплексу що реалізує сегментацію об'єктів на аерофотознімках;
- 4) дослідження шляхів застосування у бізнесі розробленої моделі та алгоритму.

1 ВВЕДЕННЯ У ПРЕДМЕТНУ ОБЛАСТЬ

Постановка задачі:

1. Провести дослідження існуючих підходів сегментації та розпізнавання об'єктів на зображеннях.
2. Вибрати підхід, який найбільш ефективно виконує задачу сегментації об'єктів на супутникових знімках.
3. Підібрати навчальну вибірку (супутникові знімки і відповідна маска об'єктів на них).
4. Реалізувати програмний комплекс, що реалізував автоматичну розмітку супутникових знімків.

1.1 Актуальність

З найдавніших часів перед людством стояла проблема орієнтування на місцевості, хоча ізольованим громадам цілком вистачало природних орієнтирів, на кшталт великих дерев або каменів. Однак, по мірі зростання поселень і встановлення зв'язків з іншими громадами почали виникати потреби в подорожах будь то торгівля, дипломатія або військові походи, що загостило проблему пошуку оптимального шляху в точку призначення і дало поштовх до появи і розвитку картографії. Перші карти описували великі і легко помітні природні об'єкти, поселення, проте були досить умовними і не точними хоча і істотно допомагали з орієнтуванням на місцевості і були бажаним придбанням як для купців так і для полководців. Згодом, завдяки розвитку торгівлі, карти

ставали все більш якісними - на них наносилися дороги, будівлі, височини, течії.

Однак, по мірі розвитку цивілізації розкрилася нова проблема картографії - старіння карт. Оскільки людина любить жити в комфорті, вона активно змінює навколишню місцевість в своїх цілях: будує нові дороги, осушує болота, вирубує ліси, зводить дамби. Таким чином, досить точні карти стають застарілими і неточними внаслідок діяльності людини чи деяких природних катаклізмів на кшталт виверження вулканів, лісових пожеж або землетрусів. Щоб краще зрозуміти суть проблеми варто розглянути зразки древніх і сучасних карт. Для демонстрації розбіжностей було вирішено взяти карту нашого краю складену французьким вченим приблизно в XVII столітті (рисунок 1.1), і сучасну карту надану сервісом Google.



Рисунок 1.1 - Фрагмент карти XVII століття

На зазначеній вище стародавній карті зображено фрагмент Південної частини України, починаючи від міста Черкаси до гирла річки Дніпро, на неї нанесені великі міста і фортеці, а також ліси, річки і кордони держав. Дана карта була досить точна для свого часу і могла послужити хорошою підмогою для

мандрівника, котрий подорожує, наприклад з міста Черкаси в місто Чигирин, і хоча обидва міста знаходяться досить недалеко, мандрівник міг би орієнтуватися на Дніпро, ліси і річку Тясмин, яка роздвоюється неподалік від Дніпра, пройшовши місто Чигирин. Таким чином, орієнтуючись на цю карту мандрівник міг дійти до потрібного йому міста уникнувши блукань по околицях.

Однак, чи зможе скористатися даною карткою і досягти точки призначення наш сучасник? Щоб відповісти на це питання потрібно поглянути на зразок сучасної карти (Рисунок 1.2), оцінити відмінності, які з'явилися за триста років.

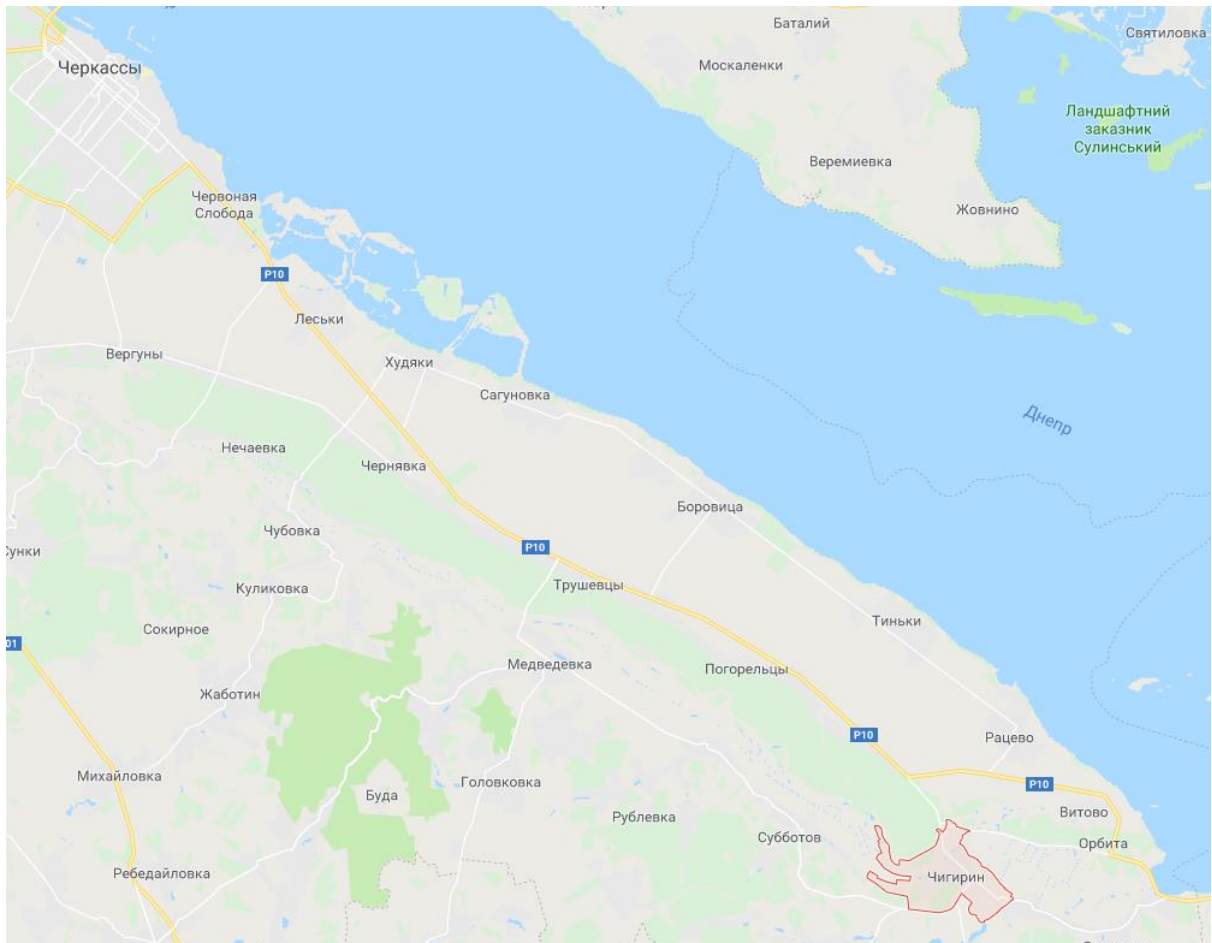


Рисунок 1.2 - Сучасна карта

Перше що кидається в очі при перегляді сучасної карти це відсутність річки, котра протікає крізь Чигирин, яка роздвоювалась перетинаючи місто і

потрапляла в Дніпро. Також видно, що місто зміщене ближче до Дніпра, який в свою чергу теж змінив свої обриси, крім того по шляху від міста Черкаси до міста Чигирин з'явилося багато поселень, доріг і полів (рисунок 1.3). Отже, чи зможе мандрівник спираючись на карту трьохсотрічної давності потрапити з Черкас до Чигирина?



Рисунок 1.3 - Східні околиці міста Чигирин

Безумовно - зможе, але наскільки ефективним буде його шлях? Адже навіть найголовніший орієнтир - Дніпро істотно змінився. Щоб дістатися до пункту призначення нашому мандрівникові потрібно буде йти вздовж річки і шукати маленьку роздвоєну річку, проте він її не знайде, оскільки на сьогоднішній день вона не роздвоюється і, як видно, з рисунка 1.3 обросла полями і іригаційними каналами, так що напевно наш мандрівник пройде повз. У кращому випадку, збитий з пантелику мандрівник зможе запитати шлях в одному з сіл, повз які він проходив, саме тому наш мандрівник зможе досягти мети, нехай і втративши час.

Однак, те що прийнятним для простого мандрівника не буде нормою, для торгівлі та логістики адже, подібні проблеми можуть коштувати не тільки часу, що само по собі дуже важливо в справі вантажоперевезень, а ще й витрат

на паливо, обслуговування транспорту, підтримання вантажу в товарному стані, особливо якщо вантаж має властивість швидко псуватися. Таким чином застарілі карти зовсім не підходять для промислового використання в сучасному світі.

І хоча великі зміни ландшафту відбуваються досить рідко, і зазвичай наносяться на сучасні карти цього все-таки недостатньо для сучасного світу, де час вирішує все. Однак, як в даному випадку може допомогти автоматичне розпізнавання об'єктів на карті, якщо є вже досить точні супутникові карти(рисунк 1.4) і знімки? Хіба супутникових знімків недостатньо щоб точно орієнтуватися на місцевості? Адже вони робляться досить часто і дуже актуальні.



Рисунок 1.4 - Поля та іригаційні канали

У випадку з мандрівником, який може бути абсолютно не знайомий з околицями, автоматична система може розпізнати стежки, поля дороги і будівлі які не можуть зацікавити користувача, чим допоможе прокласти оптимальний маршрут, так наприклад, простий мандрівник шукає готель, а далекобійник - склад, завод або автозаправку, крім того деякі об'єкти на супутниковому знімку можуть здаватися тим чим вони не є. Так, наприклад, незнайомий з місцевістю користувач може прийняти межі між полями на Рисунку 1.4 за лісо-смуги або ґрунтові дороги, однак якщо подивитися ближче, то стане очевидно, що це - різного ступеня сухості іригаційні канали, що в кращому випадку може збити з пантелику, а в гіршому втопити транспортний засіб мандрівника в одному з них.

Хоча картографія це перше що приходить на думку при роздумах про розпізнавання об'єктів на супутникових знімках, це не єдина можливість застосувати сучасні технології для поліпшення господарства і життя людини (рисунок 1.5).

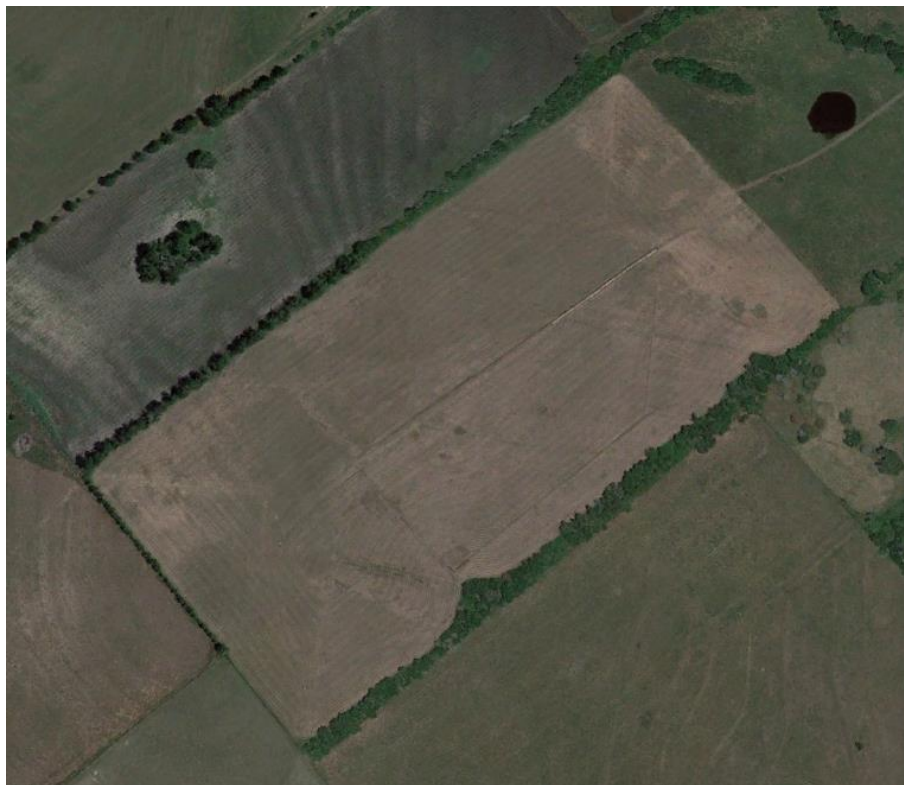


Рисунок 1.5 - Недавно засіяне поле

Варто окремо винести можливість моніторингу різного роду посівів сільськогосподарських культур і лісових масивів. Так, наприклад на рисунку 1.5 зображено поле, на якому зовсім недавно проводилися посівні роботи, а саме в травні. На малюнку помітні сліди обробки поля за допомогою сільськогосподарської техніки, однак, сходи посіяних культур все ще непомітні.

На рисунку 1.6 зображено те ж саме поле, що було раніше представлено на рисунку 1.5, проте через місяць.



Рисунок 1.6 - Поле з пророслими культурами

Як видно, різниця дуже суттєва - сліди сільськогосподарської техніки зникли під пророслими зеленими посівами. З огляду на ступінь відмінностей між господарськими етапами, визначення ступеня зрілості посівів цілком до-

ступно засобів машинного навчання. Так, наприклад, якщо буде створена модель здатна оцінювати ступінь дозрівання або стану посівів за даними аерофотозйомки, а також необхідна інфраструктура (аеростати, дрони для отримання самих знімків), то можна буде вести постійний автоматизований моніторинг величезних площ без витрат на транспорт для доставки фахівців. Крім визначення ступеня зрілості даний підхід допоможе виявити і оцінити ушкодження посівів нанесених різного роду захворюваннями, погодними явищами або шкідниками.

Варто виділити такі напрямки в картографії як екстрена і військова. Потреба в екстреній картографії виникає під час різного роду надзвичайних подій та при ліквідації наслідків. До таких прищестя можна віднести такі стихійні та техногенні лиха як:

- землетруси з масовими руйнуваннями або виникненням провалів в землі;
- повені, які зачіпають населені пункти і господарську інфраструктуру;
- урагани і смерчі призводять до знищення будівель і спонтанного переміщення великих тіл (автотранспорт ітп.);
- великомасштабні пожежі в житловій місцевості або лісових масивів;
- техногенні катастрофи (викиди шкідливих речовин заводами, аварії потягів, транспортних кораблів) котрі можуть спричинити помітний на аерофотознімках ефект (дивного кольору хмари, нафтові плями).

Що стосується військової картографії, то її специфіку також можна віднести до екстреної, оскільки бойові дії характеризуються непередбачуваністю та суттєвими змінами ландшафту, котрі часто викликані наступними факторами:

- фортифікаційні роботи;
- руйнування внаслідок обстрілів;
- мінні поля.

І хоча деякі з цих факторів розпізнати за допомогою аерофотознімків практично неможливо (мінні поля), то інші досить добре піддаються розпізнаванню та моніторингу.

На рисунку 1.7 зображений супутниковий знімок лісової пожежі, як можна помітити область знаходиться в процесі горіння і все-ще не порушена істотно розрізняються між собою і можуть бути легко знайдені, ідентифіковані та окреслені за допомогою навченої моделі.

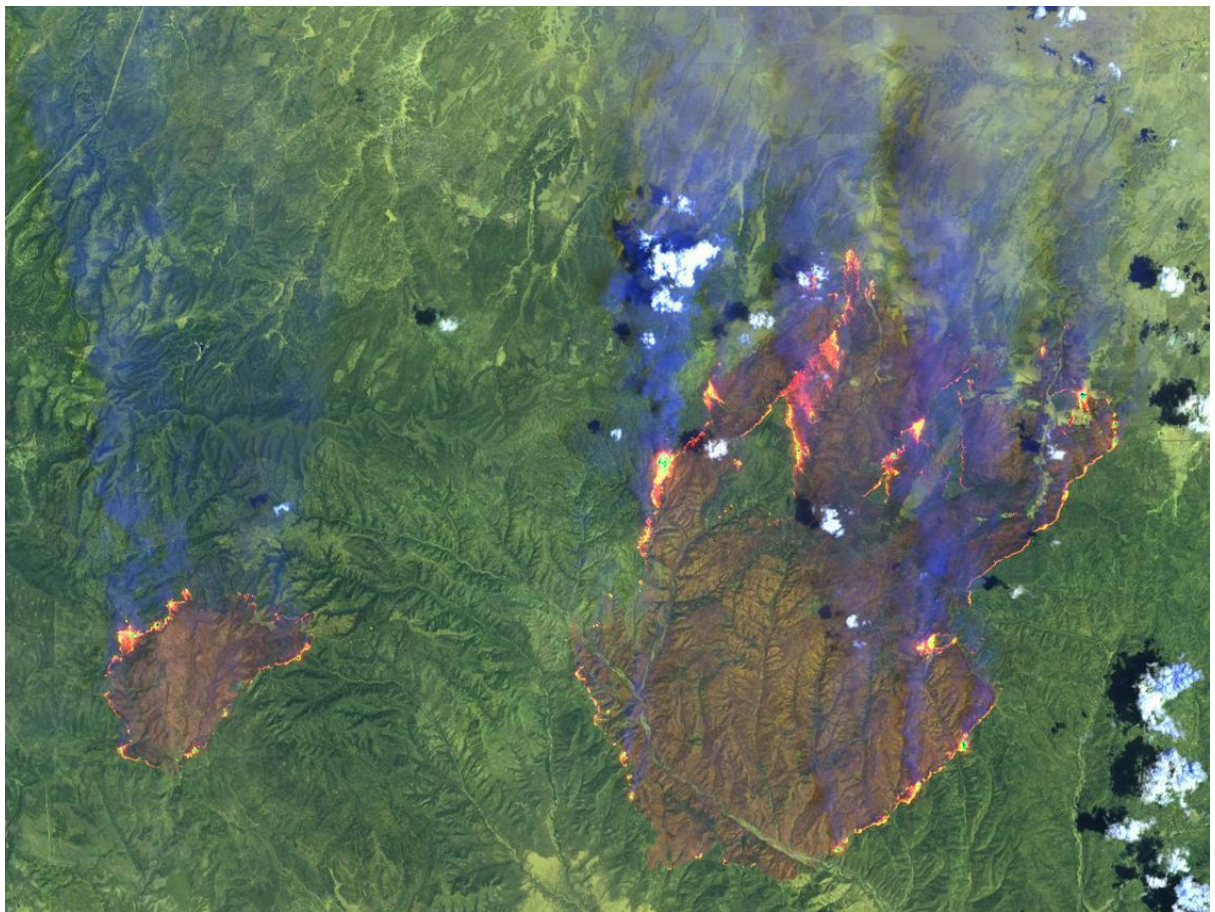


Рисунок 1.7 - Лісова пожежа

Завдяки автоматичному, динамічному моніторингу можна оперативно

помічати і наносити на карту подібні явища, що безсумнівно допоможе при процесі гасіння, захисту мешкаючих неподалік громадян, евакуації та усунення наслідків таких стихійних лих.

Рисунок 1.8 дає зрозуміти що деякі природні лиха істотно змінюють ландшафт і здатні істотно перешкодити орієнтування на місцевості, і що найгірше, процесу пошуку і порятунку потерпілих, особливо з огляду на ступінь руйнування інфраструктури.

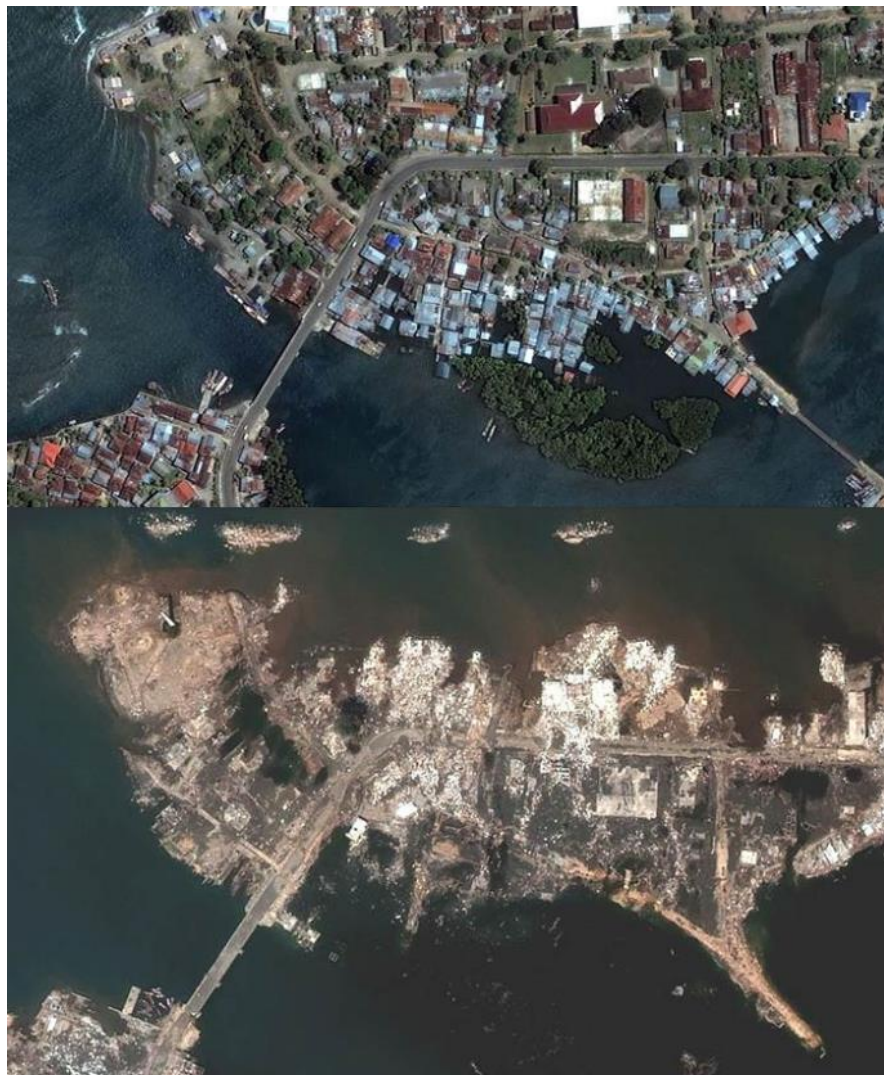


Рисунок 1.8 - Наслідки цунамі

В даному випадку модель здатна буде визначити придатні до використання транспортні шляхи, виявити завали, транспорт що перекриває шлях ітд.

Однак, на ландшафт впливають не тільки природа і промисловість але і бойові дії, також військове командування є одним з найдавніших споживачів послуг картографів, оскільки від ландшафту, розташування будівель, і правильно побудованого маршруту часто залежить успіх бойових дій. Так, на жаль, нашій власній армії довелося повною мірою відчувати на собі проблеми пов'язані з застарілими картами місцевості, на перших етапах проведення операції в зоні АТО.

На наведеному вище рисунку 1.9 зображений фрагмент досить якісної і деталізованої військової карти населеного пункту Костянтинівка, в Донецькій області.

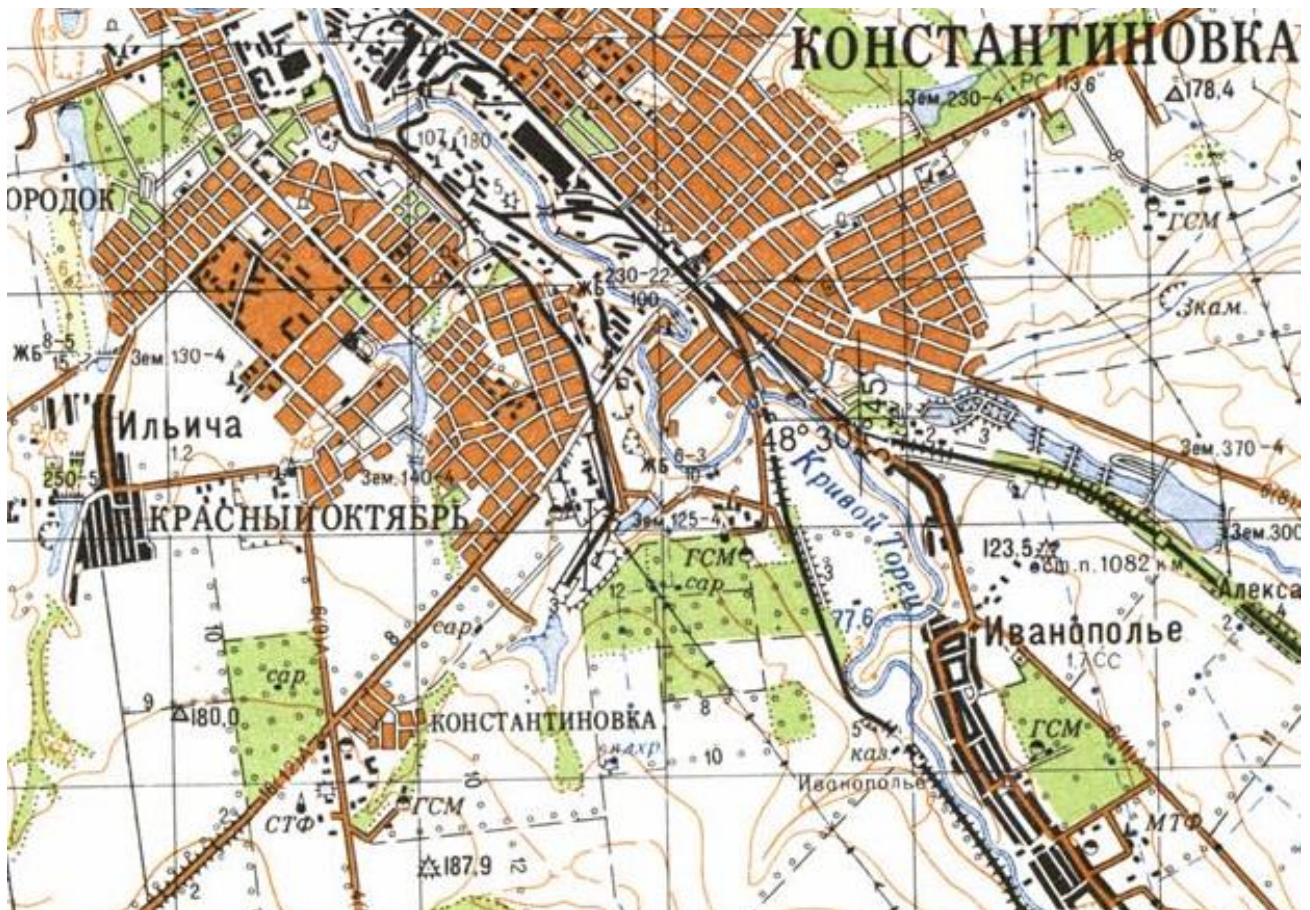


Рисунок 1.9 - Фрагмент військової карти 1985 року

З огляду на той факт що ми не маємо доступу до сучасних військових

карт даної місцевості, втім як і наші військові під час початку кампанії в 2014 році, порівняння буде відбуватися з картами наданими сервісом Google maps. Для подальшого порівняння карт, давайте звернемо увагу на посадку, яка знаходиться трохи південніше селища Червоний жовтень.

Дивлячись на наведений вище рисунок 1.10 можна помітити що багато чого змінилося за минулі десятиліття: зник Червоний жовтень - перетворився на промисловий комплекс, посадку викорчували і перетворили в сільськогосподарські землі, по суті у відкрите поле, і це не рахуючи безліч інших міnorних змін таких як перейменування і зміни структури поселень. Зміни занадто значні, аби їх можна було б ігнорувати.

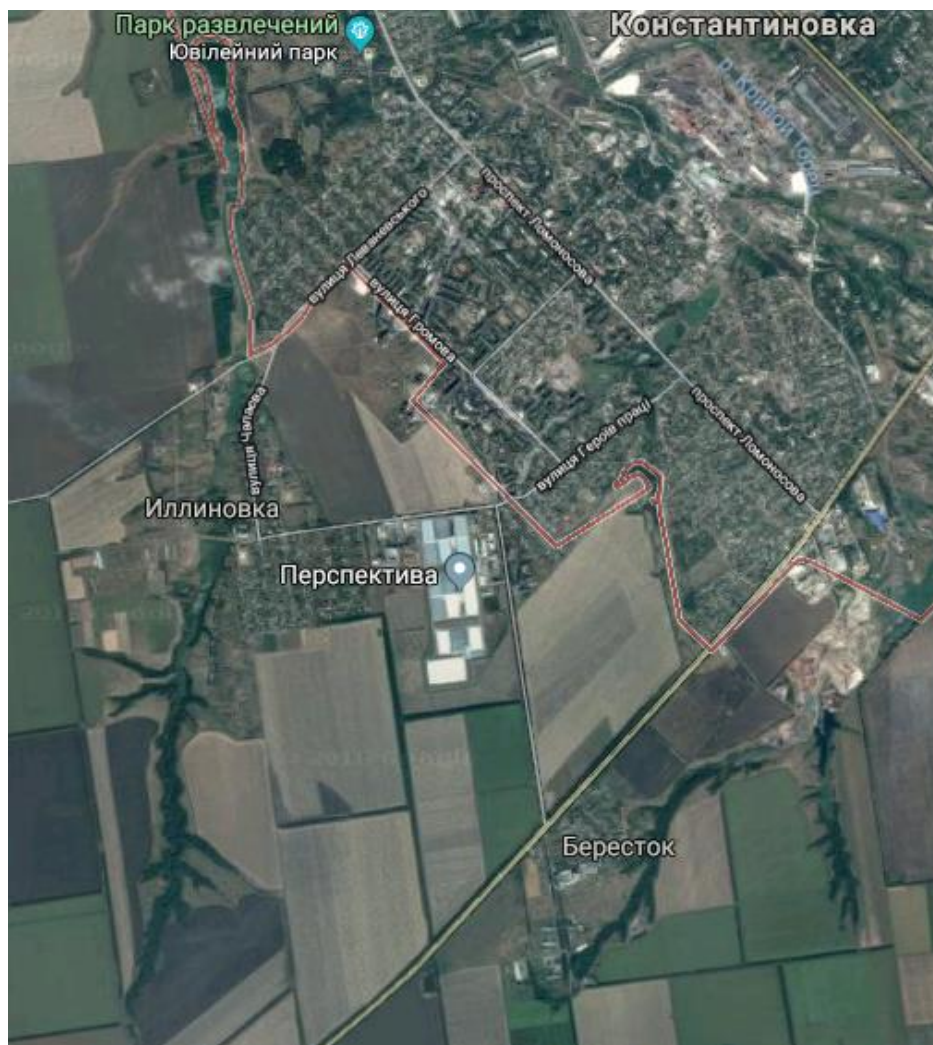


Рисунок 1.10 - Сучасна Костянтинівка

Очевидно, що навіть для звичайного мандрівника такі зміни можуть стати причиною блукань і невизначеності, а для військових, під час проведення бойової операції - тим більше, адже крім банальної можливості заблукати, в пошуках якогось селища "Червоний жовтень" військові командири могли розраховувати на використання посадки як укриття для розвідників або як джерело будівельних матеріалів для укріплень на місці, однак замість посадки зараз знаходиться поле, а замість селища - промислова зона. Подібні фактори суттєво ускладнюють всі аспекти ведення бойових дій від логістики та розвідки до артилерійських дуелей. Застарілі карти, очевидно, коштували нервів нашим військовослужбовцям, а іноді навіть життя і проваленого завдання.

Завдання картографії в цілому, і розпізнавання об'єктів на супутникових знімках зокрема актуальна як ніколи раніше. Правильно складені карти, з детально розписаними і розпізнаними об'єктами здатні не тільки економити час мандрівникам і перевізникам, а й рятувати життя у випадках природних, техногенних катастроф або військових дій. Також розпізнавання об'єктів здатне допомогти фермерам і контролюючим органам збирати відомості про стан посівів та лісових масивів.

1.2 Системи зондування земної поверхні

Великі неточності древніх карт можна пояснити відсутністю у картографів багатьох інструментів з поміж тих які в повній мірі доступні сучасним картографам. Головним з таких інструментом є можливість вести спостереження за рельєфом місцевості під потрібними кутами і з великої висоти. Картографи двісті років тому і мріяти не могли про такі можливості, тому й складали карти як бачили, що безсумнівно призводило до помилок. Однак, все перевернула

поява аерофотозйомки, яка і дала можливість отримувати достовірні знімки місцевості і на їх основі складати карти.

На рисунку 1.11 зображено перший знімок з повітря, на якому зображено місто Париж, зроблений він в 1858 році Гаспаром-Феліксом Турнашоном з використанням повітряної кулі.



Рисунок 1.11 - Перший знімок з аеростата

Згодом була розроблена спеціалізована апаратура для ведення аерофотозйомки, і навіть спеціалізовані аеростати, літаки і дрони.

У наші дні найбільш активно для аерофотозйомки застосовуються вертольоти, невеликі літаки. Також відбувається активний перехід на спеціалізовані дрони завдяки зростаючій ефективності та доступності даного виду засо-

бів аерофотозйомки. Давайте порівняємо перший знімок земної поверхні з аеростата з сучасним знімком поверхні землі (рисунок 1.12).



Рисунок 1.12 - Сучасна аерофотозйомка

Сучасні аерофотознімки на багато якісніше ніж найперші представники даного класу, за якістю вони нічим не поступаються супутниковим, крім хіба що того факту, що супутник здатний охопити великі площі. Тим не менше, варто відзначити той факт, що на теренах нашої держави необхідна для роботи безпілотних дронів інфраструктура зовсім не розвинена, і знаходиться в зародковому стані. Варто відзначити, що за час існування аерофотозйомка зазнала суттєвих змін, в тому числі виникли певні напрямки в проведенні подібного роду робіт. Давайте розглянемо два основних види аерофотозйомки, а саме: плановий (рисунок 1.13) і перспективний.



Рисунок 1.13 - Планова аерофотозйомка

Почнемо з планової аерофотозйомки. Даний вид зйомки є найбільш популярним, оскільки дозволяє в точності передати розташування об'єктів на місцевості з мінімальними неточностями і помилками залежними від кута під яким ведеться зйомка. Як неважко здогадатися вказаний вище знімок був зроблений під прямим кутом. Найбільш популярний даний вид зйомки у картографів, особливо у військових картографів, оскільки він дозволяє максимально точно зобразити техногенні та природні об'єкти на місцевості, так само даний вид зйомки часто використовують в геодезії, розмітці сільськогосподарських угідь, контролю лісових насаджень. В цілому і загальному даний спосіб характерний для військового і промислового застосування.

Рисунок 1.14 показує нам типовий приклад такого виду аерофотозйомки як панорамна зйомка. На відміну від попереднього способу тут важлива не стільки точність у визначенні місцезнаходження об'єктів скільки прагнення передати красу і вид об'єктів з боку, з висоти пташиного польоту.



Рисунок 1.14 - Панорамна аерофотозйомка

Даний вид аерофотозйомки часто знаходить своє застосування в споживчому секторі, оскільки створює дійсно красиві зображення які можуть бути використані в якості прикрас або реклами. Також панорамна аерофотозйомка може, і часто доповнює планову, оскільки деякі об'єкти буває важко помітити або описати під прямим кутом, в таких випадках може допомогти зйомка під іншими кутами, з різних ракурсів.

І хоча такий засіб дистанційного зондування землі як аерофотозйомка існує і активно працює вже більше ста років, безсумнівно варто згадати і винести на розгляд такий новий і надзвичайно ефективний спосіб зондування і картографування як використання різного роду штучних супутників землі і космічних об'єктів здатних нести фотоапаратуру (рисунок 1.15).

Всього існує два види супутників, це геостационарні і полярно-орбітальні супутники, всі вони здатні виконувати цілий спектр завдань який не входить в предметну область даної роботи.



Рисунок 1.15 - Супутниковий знімок

Однак одне із завдань даних апаратів нам дуже до речі, бо вони цілком пристосовані для отримання інформації про властивості наземних об'єктів і явищ шляхом реєстрації вихідного від них електромагнітного випромінювання. Встановлена на супутниках апаратура для дистанційного зондування землі може працювати в наступних основних чотирьох діапазонах [1]:

- ультрафіолетовий діапазон;
- видимий діапазон;
- інфрачервоний діапазон;
- мікрохвильовий діапазон.

У видимому діапазоні знімальна апаратура супутника реєструє відбите від земної поверхні і пройшовши крізь атмосферу сонячне випромінювання. У мікрохвильовому діапазоні використовується власне відображення поверхні нашої планети, або антропогенний випромінювання спрямованої не поверхня

землі з метою отримання даних. Що стосується інфрачервоного діапазону (рисунк 1.16), то тут використовується власне теплове випромінювання землі. Знімки в інфрачервоному діапазоні активно застосовуються в метеорології для передбачення зміни погоди, оскільки, дозволяють визначати термодинамічні властивості повітряних мас, хмарність.

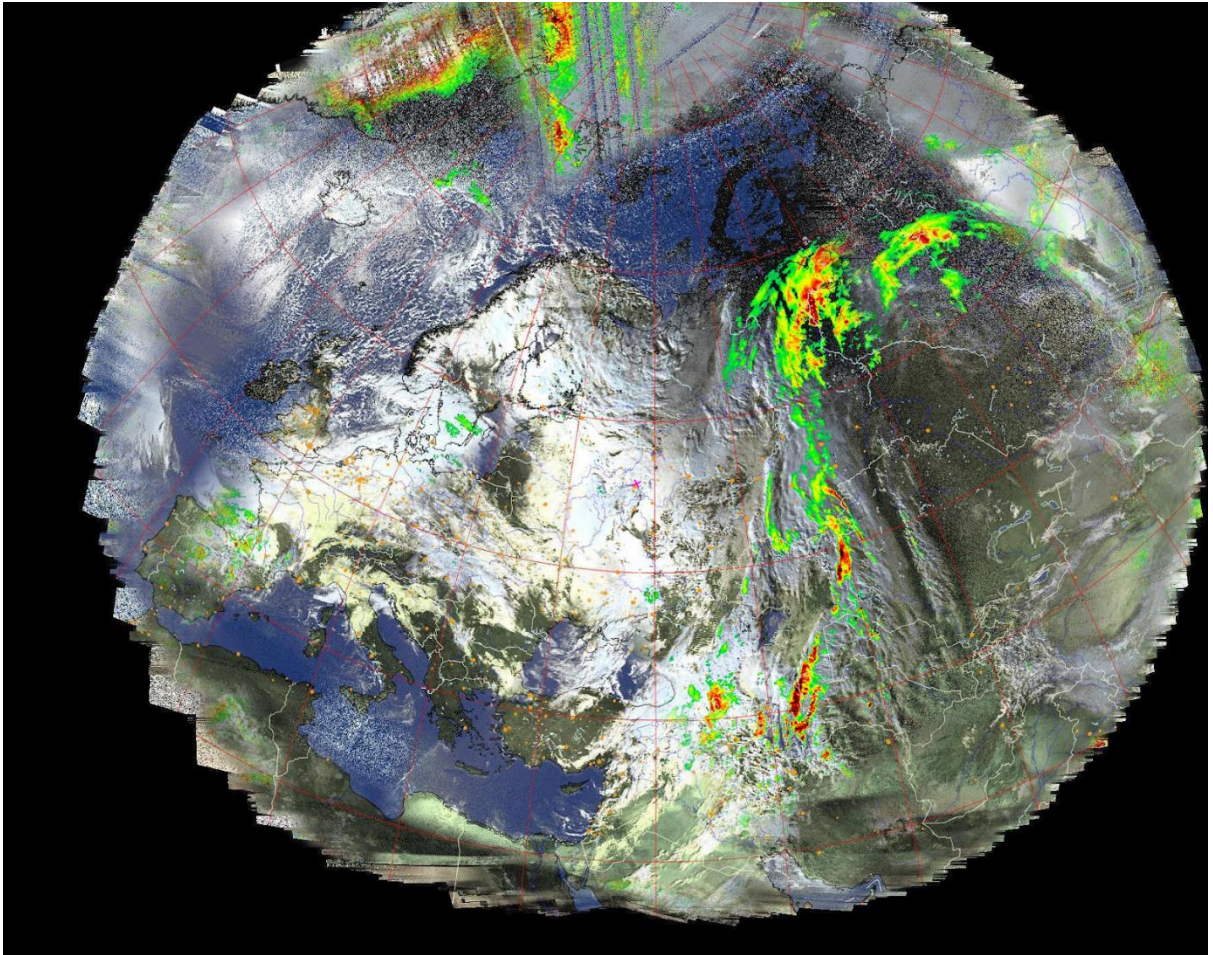


Рисунок 1.16 - Інфрачервоний супутниковий знімок hobitus

1.3 Задача сегментації зображення

Дана дослідницька робота цілком і повністю ґрунтується на використанні різного роду зображень земної поверхні, і як наслідок вимагає виділення зображених об'єктів один від одного для подальшої класифікації.

Так склалося еволюційно, що людський мозок виконує цю функцію автоматично і блискавично, аналіз роботи мозку допоможе вирішити поставлену перед нами проблему. Так, наприклад нашому мозку надійшло зображення на якому знаходиться будинок, і зображення на якому знаходиться сарай. Як визначити де будинок і де сарай? Обидва об'єкти, безумовно є будівлями, причому обидва можуть бути виготовлені з одного і того ж матеріалу дерева або цегли, і само собою можуть бути одного і того ж розміру, або різних розмірів, так як же визначити де будинок а де сарай? Для цього необхідно виділити унікальні ознаки будинку і сараю, найбільш простий спосіб зробити це - розбити дані будови на сегменти.

Рисунок 1.17 показує приклад сегментації (розбиття на сегменти) фотографії на якій зображений будинок і деякі околиці, такі як небо, автомобіль, кущі.



Рисунок 1.17 - Будівля, розділена на сегменти

Однак, давайте загостримо увагу на самому будинку: що з виділеного на зображенні може бути властиво дому і не властиво сараю? перше що впадає в очі - кількість і розмір вікон - в житловому будинку буде більше вікон і вони будуть більшого розміру, також житловий будинок як правило вище сараю, і сараї зазвичай не мають вікон на даху, двері у сараю як правило більше ніж у будинку. Ґрунтуючись на наведених вище ознаках, які можна отримати з сегментованого зображення можна з високим ступенем вірогідності правильно виконувати завдання бінарної класифікації, а саме відрізняти будинки від сараїв.

Хоча рисунок 1.18 показує велику кількість алгоритмів, існує ще безліч цікавих і ефективних рішень.

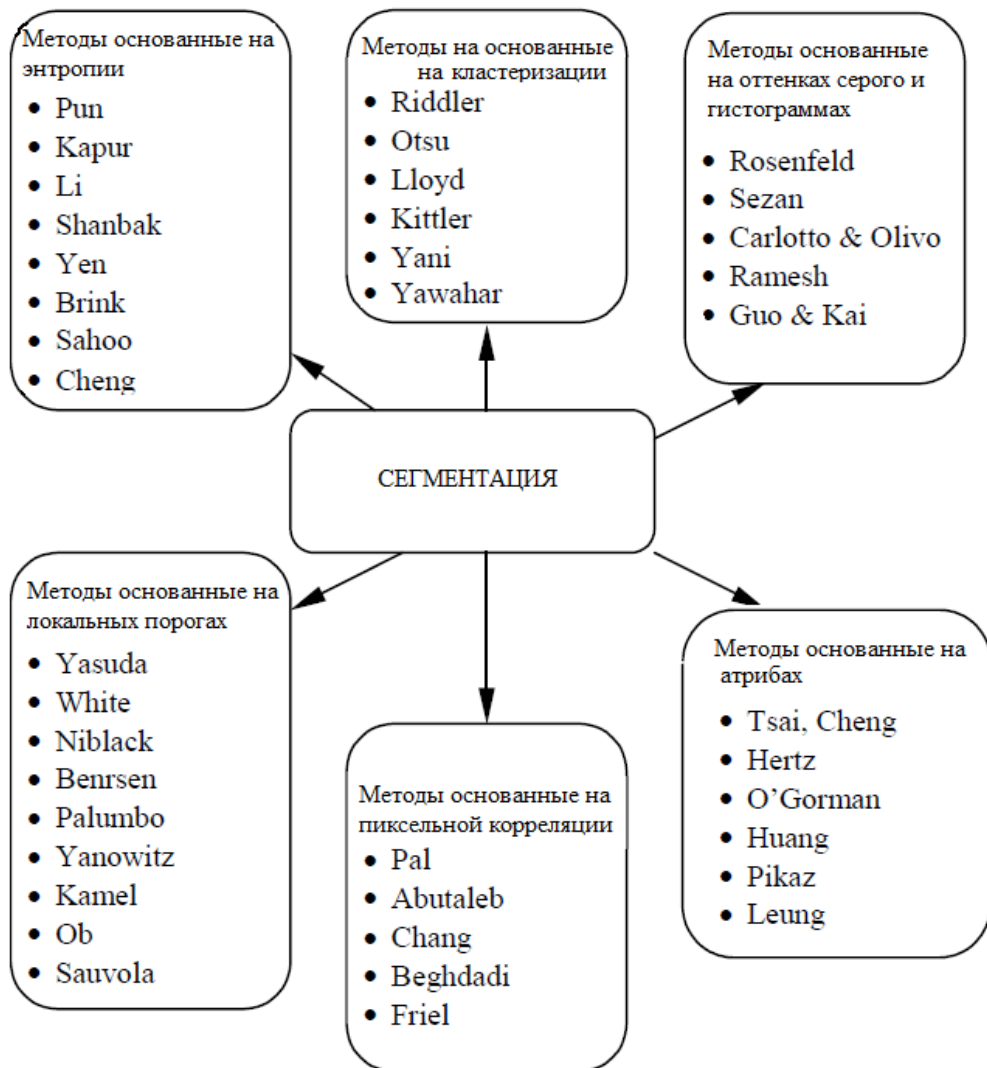


Рисунок 1.18 – Методи сегментації зображень

Давайте розглянемо алгоритм сегментації по вододілах (WatherShed). Даний алгоритм працює з функцією двох змінних.

Рисунок 1.19 показує приклад сегментації методом вододілу, як видно на малюнку було виділено два сегменти: чарівник і околиці.



Рисунок 1.19 - Приклад сегментації методом водорозділу

Що стосується методу роботи даного алгоритму то тут все досить просто: якщо по осі OZ відкласти абсолютне значення градієнта, то в місцях перепаду інтенсивності градієнта, утворюються височини, а в однорідних регіонах рівнини, після знаходження екстремумів функції починається процес заповнення "водою" який починається з найменшої серед мінімумів, як тільки рівень води досягне чергового локального максимуму почнеться і його заповнення, як тільки два басейни почнуть переливатися і спілкуватися між собою зводиться дама, щоб запобігти об'єднанню басейнів, заповнення басейнів триває до тих пір, поки всі регіони не розділені перегородками [2]. Наочно цей процес продемонстрований на наведеному нижче рисунку 1.20.

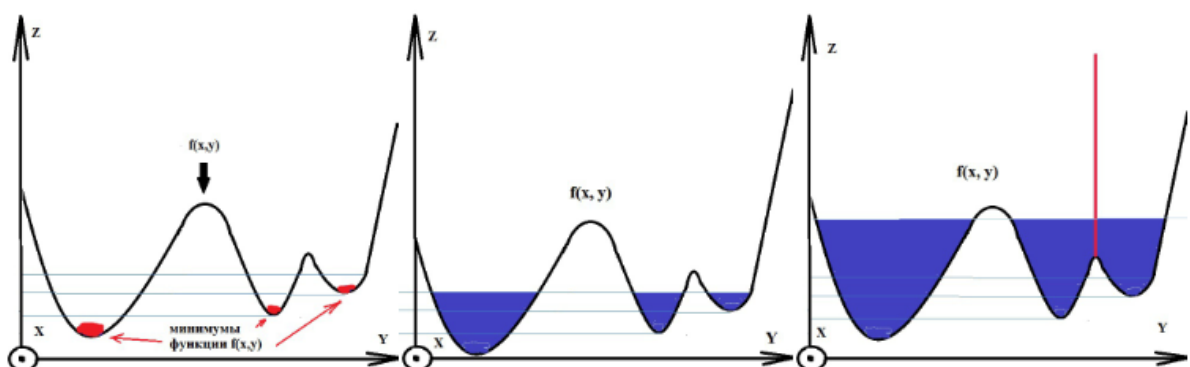


Рисунок 1.20 - Процес заповнення водою

Даний алгоритм може бути дуже ефективним при наявності невеликої кількості локальних мінімумів, проте як тільки справа доходить до великої кількості деталей даний алгоритм втрачає свою ефективність. Давайте розглянемо алгоритм сегментації MeanShift.

Алгоритм сегментації MeanShift відомий тим, що групує об'єкти зі схожими характерними ознаками в один сегмент, видаючи на виході зображення з однорідними областями [3].

Рисунок 1.21 показує приклад сегментації за допомогою алгоритму MeanShift.

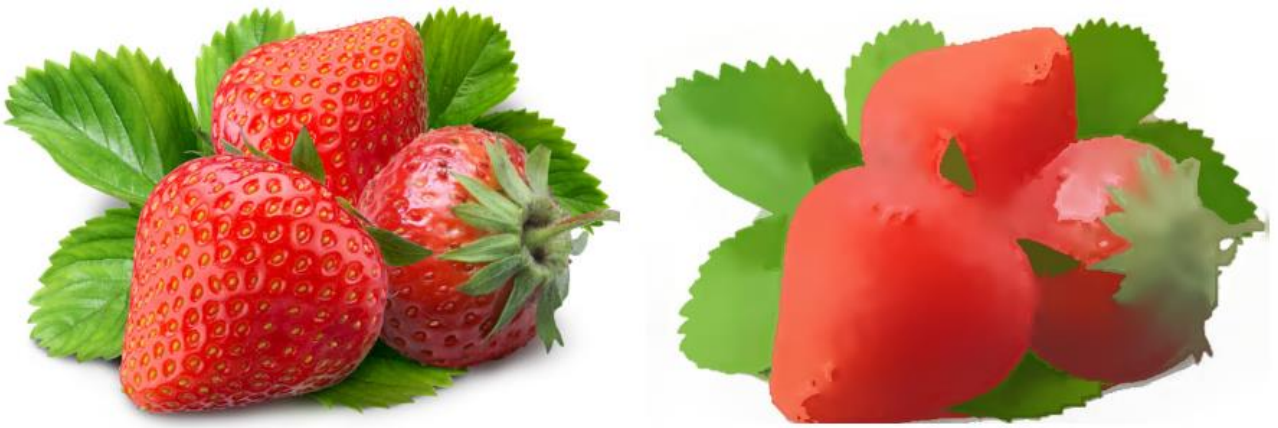


Рисунок 1.21 - Приклад роботи алгоритму MeanShift

У якості координат в просторі ознак даний алгоритм, як і попередній використовує координати пікселя x , y . Якщо зобразити пікселі в просторі ознак можна помітити згущення в певних місцях на графіку (рисунок 1.22). Також варто розглянути функцію щільності (рисунок 1.23).

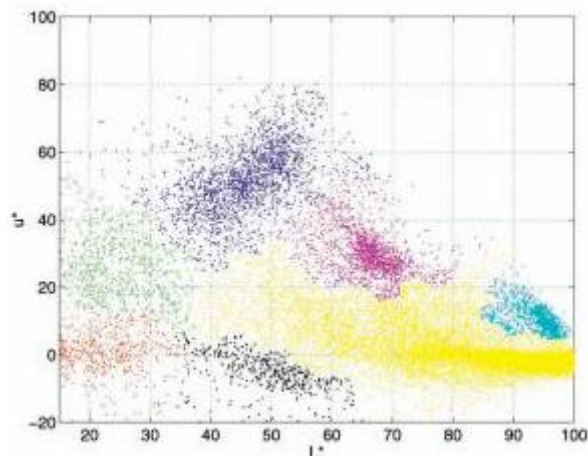


Рисунок 1.22 - Згущення пікселів в окремих місцях

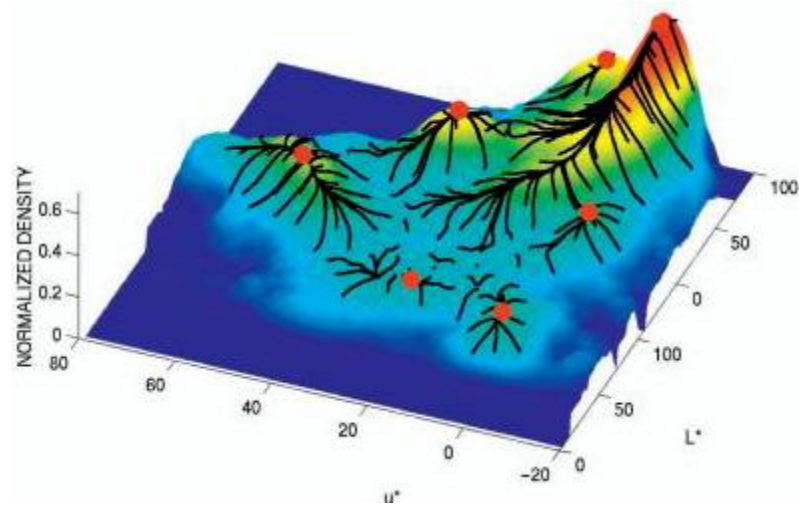


Рисунок 1.23 - Функція щільності

Функцію щільності для згущення точок можна описати так:

$$f(\vec{x}) = \frac{1}{Nh^d} \sum_{i=1}^N K\left(\frac{\vec{x} - \vec{x}_i}{h}\right) \quad (1.1)$$

де

\vec{x}_i - вектор ознак i -го пікселя,

d - кількість ознак,

N - число пікселів,

h - параметр відповідальний за гладкість.

Максимуми функції розташовані в точках згущення пікселів зображення в просторі ознак. Пікселі належать одному локальному максимуму, об'єднуються в один сегмент. Таким чином виходить, щоб знайти до якого з центрів згущення відноситься конкретно взятий піксель доводиться крокувати по градієнту для знаходження найближчого локального максимуму. Як правило вибірки координат пікселів відбувається за кількома параметрами. Так, наприклад, якщо враховувати близькість за кольором і відстані, то небо буде відібрано в один сегмент, а море в інший, в той же час якщо робити вибір тільки за колірною ознакою, то небо і море зливаються в один кластер, так як їхні координати потрапляють в один локальний максимум. MeanShift не стане об'єднувати сегменти з великою різницею в кольорах в один, в той же час якщо кольори досить однорідні, ще даний алгоритм добре справляється з однорідними кольорами на строкатому тлі, і часто застосовується для спостереження за швидкими об'єктами.

Крім даних, "строкатих" методів існують такі, які схильні до кластеризації однорідних регіонів, таких як, наприклад мікросхеми або контурні карти. Одним з таких алгоритмів є FloodFill також відомий як метод заливки або повені, з його допомогою можна виділити однорідні за кольором регіони [3]. Для цього потрібно вибрати початковий піксель і задати інтервал зміни кольору сусідніх пікселів щодо пікселя вихідного. Хоча найбільш ефективно даний алгоритм працює з несиметричними і симетричними інтервалами. Алгоритм об'єднує сусідні пікселі в один сегмент шляхом заливання цих пікселів одним кольором, якщо вони потрапляють в якийсь діапазон. На виході сегмент, залитий певним кольором і його площа в пікселях.

Такий алгоритм корисний для заливки області зі слабкими перепадами кольору з однорідним фоном. Як показано на рисунку 1.24 алгоритм FloodFill може бути виявлення пошкоджених країв об'єкта.

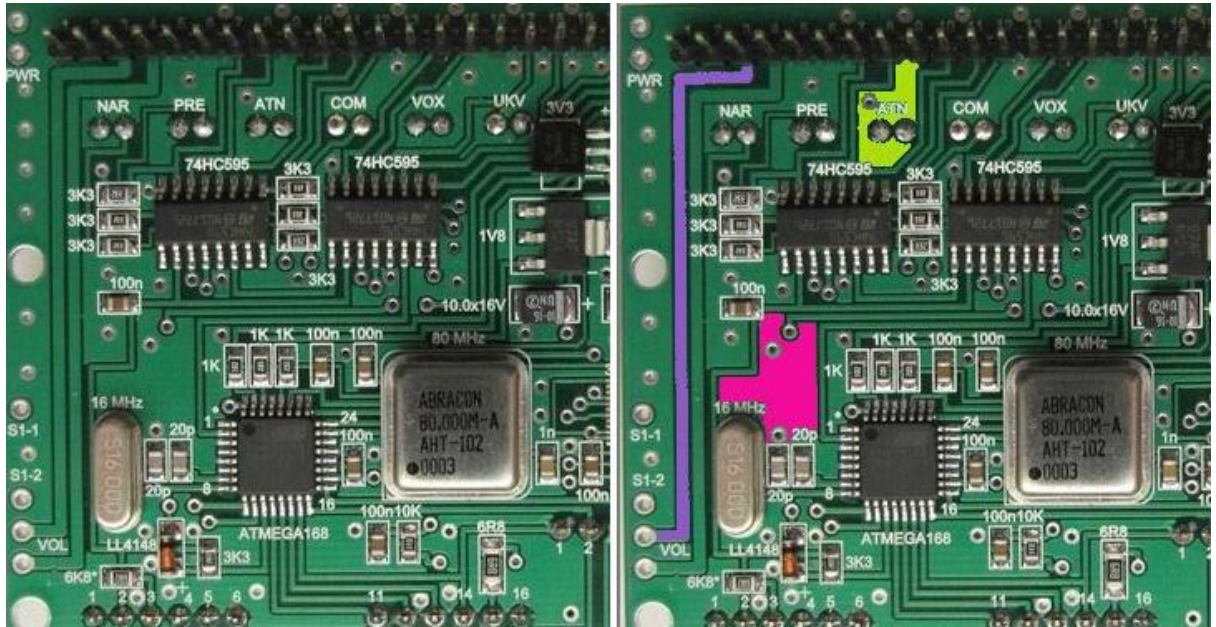


Рисунок 1.24 - Результат заливки областей

Наприклад, якщо, заливаючи однорідні області певним кольором, алгоритм заповнить і сусідні регіони, то значить порушена цілісність кордону між даними областями. На малюнку зазначеному вище цілісність кордонів зберігається.

Тепер давайте розглянемо інший алгоритм сегментації GrabCut. Цей алгоритм відноситься до інтерактивних алгоритмів виділення об'єкта, який розроблявся як альтернатива магнітному ласо. Для роботи алгоритму досить укласти об'єкт разом з частиною фону в прямокутник (рисунок 1.25).

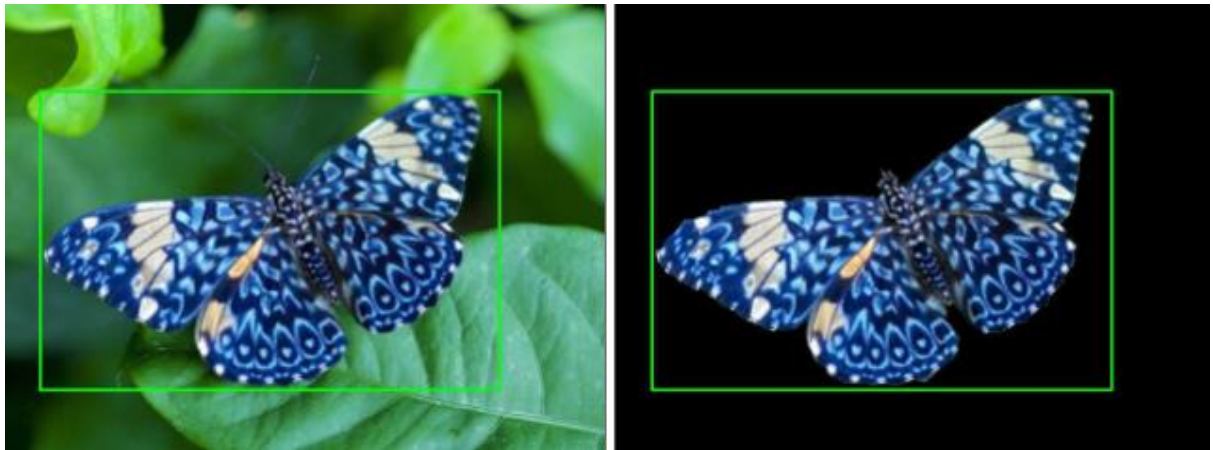


Рисунок 1.25 - Алгоритм GrabCut

Можуть виникнути складнощі при сегментації, якщо всередині прямокутника присутні кольори, котрі зустрічаються у великій кількості не тільки на цільовому об'єкті але і на всій картинці. У такому випадку прийнято ставити додаткові мітки об'єкта і фону щоб краще орієнтуватися.

Розглянемо алгоритм К-середніх (K-means). Алгоритм К-середніх є найпростішим і найбільш широко використовуваним алгоритмом кластеризації. Для заданого зображення з N пікселів метою є розбиття зображення на K кластерів, де значення K може бути задано користувачем. Кластери забезпечують групування пікселів, які залежать від їх значень на зображенні, але не обов'язково від їх розташування, якщо розташування не є заданою властивістю. Нехай $X = \{x_1, x_2, \dots, x_N\}$ - безліч пікселів і нехай $V(x_i)$ означає вектор властивостей пікселя x_i . Алгоритм К-середніх включає наступні кроки:

- 1) *Ініціалізація параметрів.* Середні кожного з K кластерів ініціалізуються значеннями потенційних векторів властивостей. У класичному алгоритмі К-середніх значення кожного елемента вектора властивостей вибирається випадково з усіх можливих значень для цього елемента. Наприклад, якщо вектор властивостей заданий у формі (H, S, V) становлять H -відтінок, S -насичення, V -значення інтенсивності, перший елемент H вибирається випадково з усіх можливих значень.

- 2) *Жорстке призначення пікселів кластерам.* Тепер, коли кожен з До кластерів S_k має своє середнє μ_k , кожен піксель призначається кластеру з найближчим середнім, використовуючи функцію відстані, яка підраховує відстань між двома векторами властивостей. У початковий момент кожен піксель x_i пов'язаний з окремим кластером S_k .
- 3) *Перерахунок параметрів.* Середні кластерів перераховуються на основі векторів властивостей всіх пікселів в кожному кластері. Таким чином величина μ_k підраховується як середнє (центр ваги) $\{V(x_i) | x_i \in S_k\}$.

Кроки 2 і 3 повторюються, поки не буде досягнута збіжність, яка настане, коли жоден піксель не перейде з одного кластера в інший на черговій ітерації. На рисунку 1.26 приведена сегментація за методом К-середніх кольорового зображення на 4 кластери. Зауважимо, що дах будівлі і поверхня землі, по якій йдуть люди, мають приблизно однаковий колір на зображенні, і вони обидва приписуються одному кластеру.

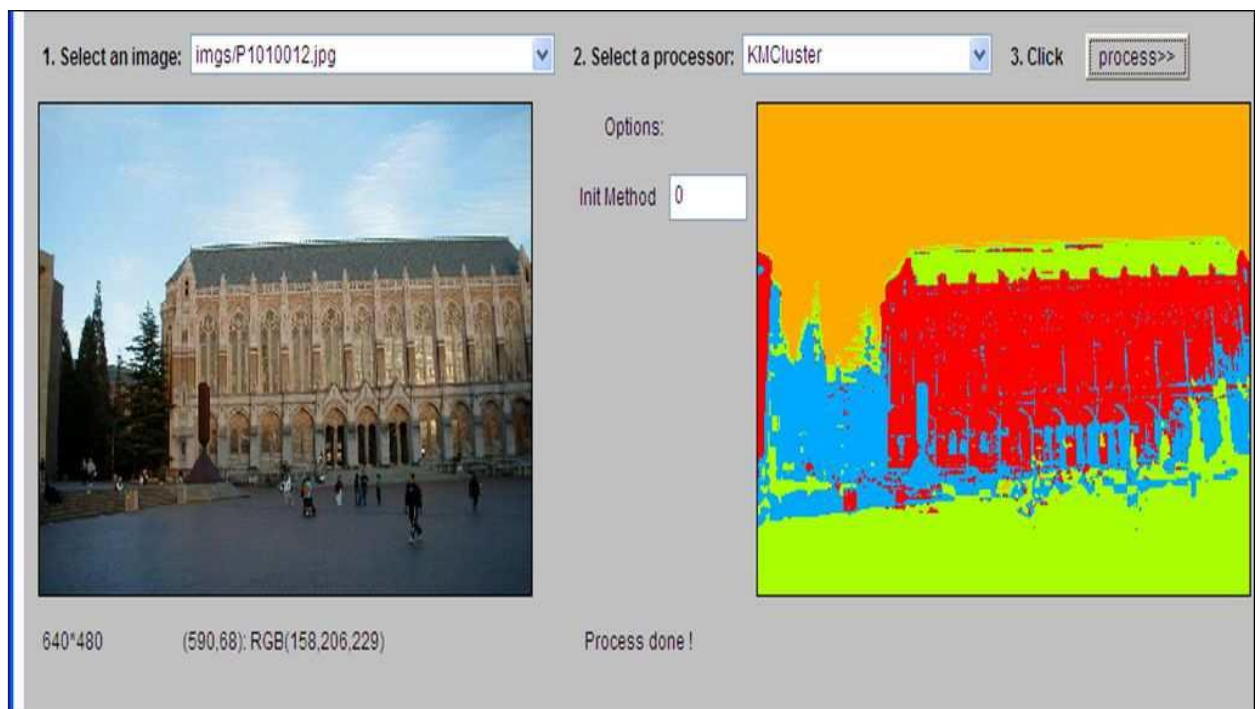


Рисунок 1.26 – Сегментація методом K-means

Розглянемо алгоритм на основі максимізації математичного очікування. Алгоритм максимізації математичного очікування (ЕМ) відповідає алгоритму К-середніх в тому, що він також вимагає, щоб користувач вибрав число кластерів і має ті ж самі 3 кроки: ініціалізація, призначення пікселів кластерам і перерахунок параметрів [4]. Це не алгоритм кластеризації, а загальний алгоритм методу, який шукає оцінки максимуму правдоподібності в параметричних моделях з неповними даними та має багато додатків, включаючи як кластеризацію, так і класифікацію.

Кластери представляються своїми розподілами ймовірностей, а не середніми. Найчастіше використовуються гауссовський розподілу. Кластери тоді представляються своїми середніми μ і матрицями коваріацій Σ . Кожен кластер має вагу w , і як тільки кластери (звані компонентами в термінах методу ЕМ) будуть підраховані, кожен вектор властивостей може бути виражений як зважена суміш гауссовський розподілів.

Призначення пікселів до кластерів є «м'яким» або імовірнісним (замість жорсткого призначення кожного пікселя точно одного класу в методі К-середніх). Кожен піксель має кінцеву ймовірність приналежності одному з кластерів. Хоча зазвичай прийнято призначати піксель кластеру з максимальною вірогідністю, можливо також зберігати повний вектор ймовірностей і використовувати його в подальшому аналізі.

Алгоритм працює наступним чином:

- 1) Ініціалізація параметрів. Ініціалізувалися параметри кожного кластера S_k - середнє (мат. Очікування) μ_k , матриця коваріацій Σ_k , і вага $w_k = p(S_k)$. Середні можуть бути ініційовані як випадкові вектора значень, як в методі К-середніх. Найпростіший спосіб ініціалізації матриці коваріацій - це покласти їх рівними одиничним матрицями $n \times n$. Ваги не започатковано значеннями $1/k$ для всіх кластерів.

- 2) М'яке призначення пікселів кластерам. На цьому кроці оцінюється умовна ймовірність $p(C_k | x_i)$ для кожного пікселя x_i і кожного кластера C_k . Ця умовна ймовірність підраховується згідно стандартному рівняння.

$$p(C_k | x_k) = p(x_i | C_k) * p(C_k) / p(x_i) \quad (1.2)$$

$$\text{де } p(x_i) = \sum_k p(C_k | x_i) * p(C_k) \quad (1.3)$$

де $p(C_k)$ - поточний вага вк. Якщо C_k описується Гауссовским розподілом, а $V(x_i)$ вектор властивостей пікселя x_i , то $p(x_i | C_k)$ задається

$$p(x_i | C_k) = \frac{1}{\sqrt{2\pi|\Sigma_k|}} e^{-\frac{1}{2}(V(x_i) - \mu_k)^T \Sigma_k^{-1} (V(x_i) - \mu_k)} \quad (1.4)$$

- 3) Перерахунок параметрів, максимізація. Параметри μ_k , Σ_k , w_k кожного кластера C_k тепер переобчислюють на основі значень векторів властивостей $V(x_i)$, всіх пікселів x_i і обчислених на другому кроці ймовірностей $p(C_k | x_i)$. Формули для поновлення параметрів наступні:

$$\mu_k = \frac{\sum_i p(C_k | x_i) * V(x_i)}{\sum_i p(C_k | x_i)} \quad (1.5)$$

$$\Sigma_k = \frac{\sum_i p(C_k | x_i) * (V(x_i) - \mu_k)(V(x_i) - \mu_k)^T}{\sum_i p(C_k | x_i)} \quad (1.6)$$

Результат роботи алгоритму ЕМ приведено на рисунку 1.27.

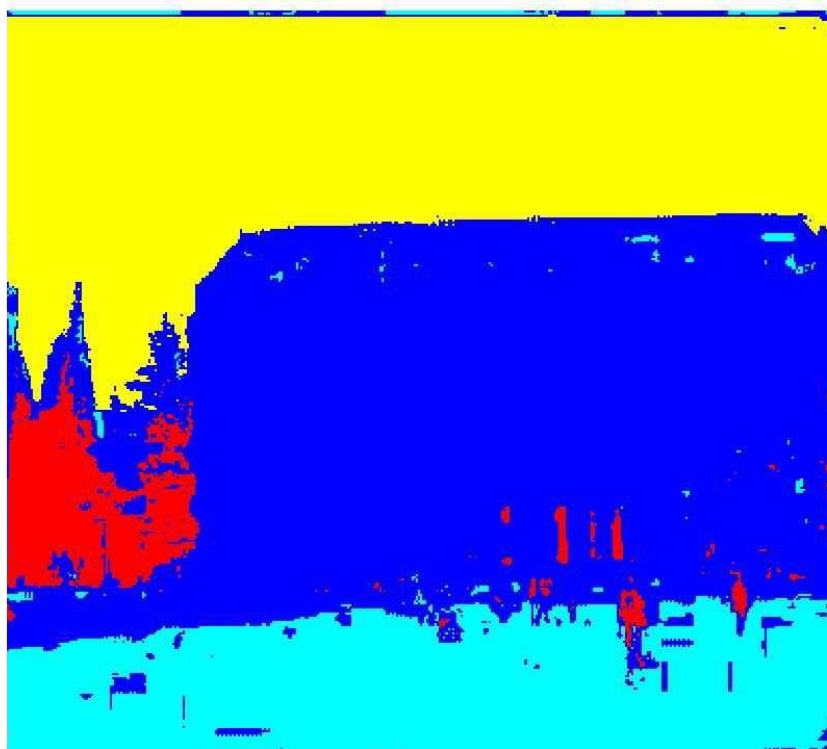


Рисунок 1.27 – Сегментація алгоритмом ЕМ

Розглянемо алгоритм сегментації на основі нормалізації перетину графів. Спектральна кластеризація - це вид кластеризації, яка використовує власні вектора даних в процедурі кластеризації. Нормалізовані перетину (розрізи) графа - це вид спектральної кластеризації, розроблений спеціально для сегментації зображень. У цьому підході пікселі зображення утворюють вузли вершини графа, чиї зважені ребра представляють подібність (в сірих тонах, в кольорі або в інших атрибутах) між пікселями, а алгоритм розрізає граф на 2 подграфа.

Нехай $G(V, E)$ - граф, чиї вершини - це точки в просторі вимірювань, а ребра мають вагу $w(i, j)$, який представляє схожість (подібність) між вершинами i та j . Мета сегментації - знайти таке розбиття графа (вершин) на окремі підмножини V_1, V_2, \dots, V_m так, щоб схожість всередині безлічі було високим, а між множинами - низьким. Граф може бути розбитий на 2 окремих подграфа з множинами вершин A і B шляхом видалення всіх ребер, які пов'язують вершини з A з вершинами з B . Ступінь відмінності («несхожості») між

множинами A і B може бути підрахована як сума ваг віддалених ребер, їх сумарна вага називається перетином (cut).

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \quad (1.7)$$

Один спосіб формування проблеми сегментації - це пошук мінімального перетину в графі і виконання цієї процедури до тих пір, поки регіони не стануть однорідними. Критерій мінімуму перетину, однак, вважає за краще вибір малих множин ізольованих вершин, який буде корисний для пошуку великих однорідних за кольором / текстурі регіонів. Нормалізоване перетин ($N-cut$) визначається в термінах перетину $cut(A, B)$ і асоціації між вершиною A і повним безліччю вершин V , яка визначається наступним чином (1.8):

$$asso(A, V) = \sum_{u \in A, t \in V} w(u, t) \quad (1.8)$$

Звідки, визначення нормалізованого перетину наступне (1.9):

$$N_{cut}(A, B) = \frac{cut(A, B)}{asso(A, V)} + \frac{cut(A, B)}{asso(B, V)} \quad (1.9)$$

При такому визначенні, перетин, яке розділяє малі ізольовані безлічі точок, не матиме малих нормалізованих перетинів $Ncut$, а розбиття, які дійсно роблять малі значення нормалізованих перетинів ($Ncut$) виявляються ймовірно більш корисними для сегментації зображень.

Процедура пошуку мінімальних нормалізованих перетинів наступна:

1) Нехай $G(V, E)$ - зважений граф, а N - розмір його безлічі вершин.

Визначимо вектор d з $d(i)$, яке задається наступним чином (1.10):

$$d(i) = \sum_j w(i, j) \quad (1.10)$$

- 2) Нехай x буде вектором, компоненти якого знаходяться наступним чином (1.11):

$$x_i = \begin{cases} 1, & \text{якщо вузол знаходиться в } A; \\ -1, & \text{в протилежному випадку.} \end{cases} \quad (1.11)$$

І нехай y буде неперервною апроксимацією x визначеною наступним чином (1.12):

$$y = (1 + x) - \frac{\sum_{x_i > 0} d_i}{\sum_{x_i < 0} d_i} (1 - x) \quad (1.12)$$

Необхідно вирішити систему рівнянь (D-W) $y = \lambda D y$ для власних векторів y і власних значень λ .

- 3) Використовувати власний вектор з другим найменшим власним значенням для розбиття графа на 2 підграфа, щоб знайти точку розбиття, для якої буде мінімізовано нормалізоване розтин.

Рисунок 1.28 ілюструє операцію алгоритму нормалізованого перетину графа на зображенні будівлі.

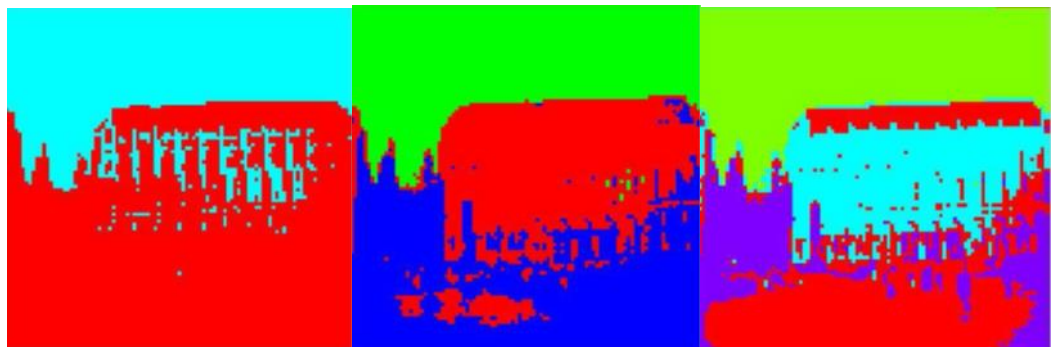


Рисунок 1.28 – Сегментація алгоритмом нормалізації перетину графу

1.4 Нейронні мережі

Нейронна мережа - це послідовність різноманітних нейронів, з'єднаних між собою синапсами [5]. Структура нейронної мережі прийшла в світ програмування з біології. Завдяки такій структурі, машина отримує можливість аналізувати і запам'ятовувати різну інформацію. Крім того нейронні мережі здатні не тільки до класифікації поданої на вхід інформації, але і здатні відтворювати її вносячи при цьому деякого роду корективи. Іншими словами штучна нейронна мережа повторює собою роботу природної нейронної мережі, яка знаходиться в мозку людини та інших біологічних видів здатних до запам'ятовування і прийняття рішень. Рисунок 1.29 показує схему роботи штучної нейронної мережі.

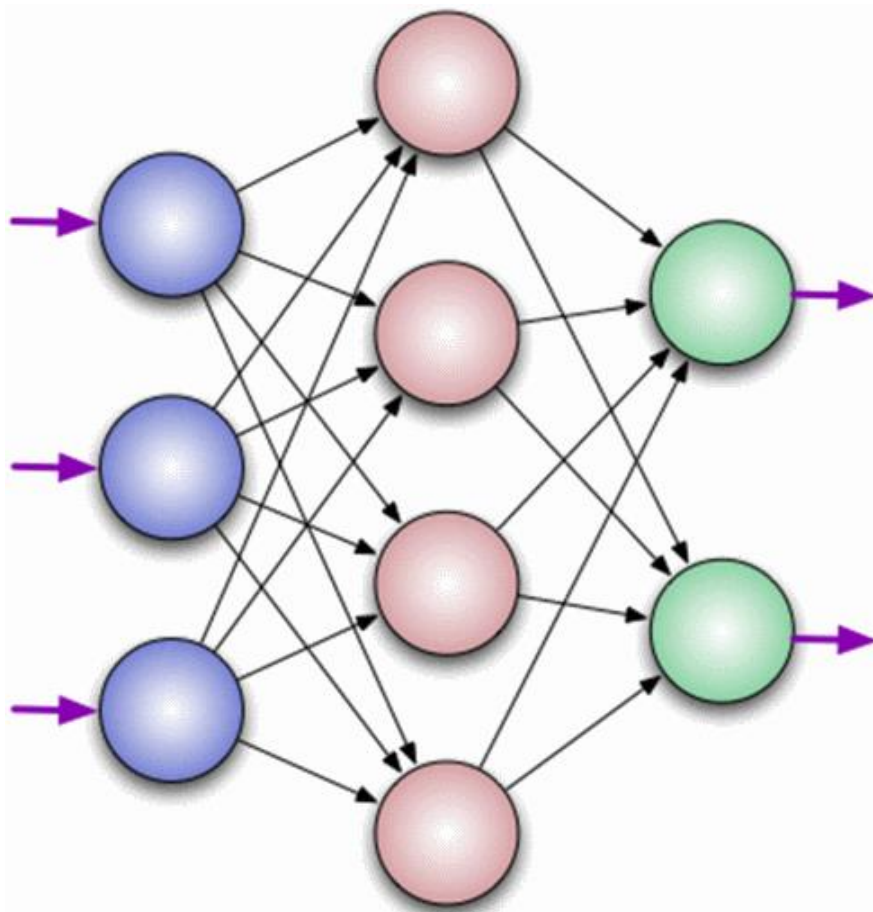


Рисунок 1.29 - Схема штучних нейронів

Сам по собі нейрон - це обчислювальна одиниця, яка отримує інформацію і виробляє над нею прості обчислення і передає далі [6]. Вони діляться на три основних типи:

- вхідний (синій);
- прихований (червоний);
- вихідний (зелений);

Також є нейрон зміщення і контекстний нейрони. У тому випадку, коли нейронна мережа складається з великої кількості нейронів застосовується термін шару. Відповідно є вхідний шар, який отримує інформацію ззовні, деяка кількість прихованих шарів, зазвичай їх близько трьох, які обробляють отриману інформацію, і вихідний шар, який і виводить результати обчислень.

Також варто розглянути синапс - це зв'язок між двома нейронами. У синапсів є тільки один параметр - вага. Завдяки йому, вхідна інформація змінюється, коли передається від одного нейрона до іншого. Припустимо, є три нейрона, які передають інформацію наступного, тоді у нас є три ваги, які відповідають кожному з цих нейронів. Той нейрон, чия вага буде найбільшою - буде домінуючим в наступному нейроні, як приклад можна привести змішування кольорів (рисунок 1.30).

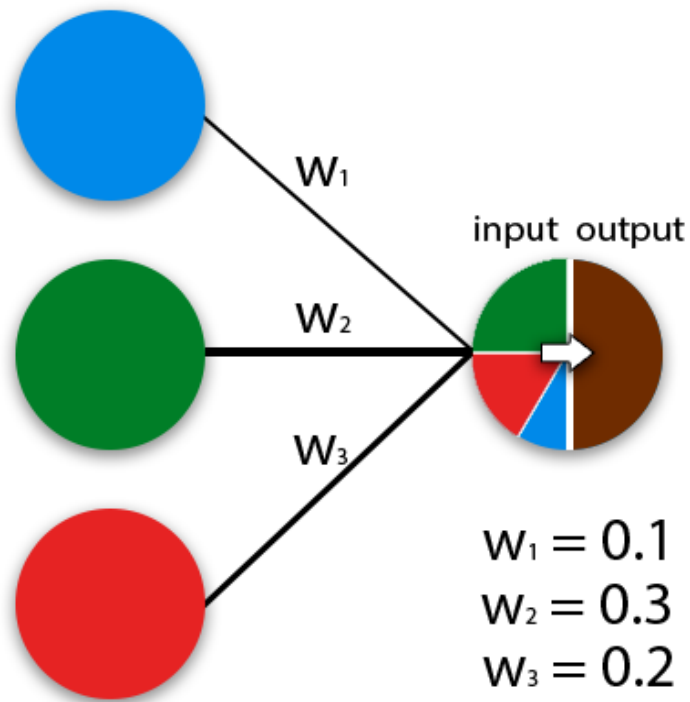


Рисунок 1.30 - Нейрон та синапси

Сукупність ваг нейронної мережі або матриці ваг - це свого роду штучний мозок системи. Саме завдяки вагам інформація обробляється і перетворюється в результат.

1.5 Згорткові нейронні мережі

Згорткова нейронна мережа складається з різних видів шарів: згорткові шари, субдискретизуючі шари і шари звичайної нейронної мережі - персептрона. Подібну структуру можна більш детально розглянути на наведеному нижче малюнку (рисунок 1.31) [7].

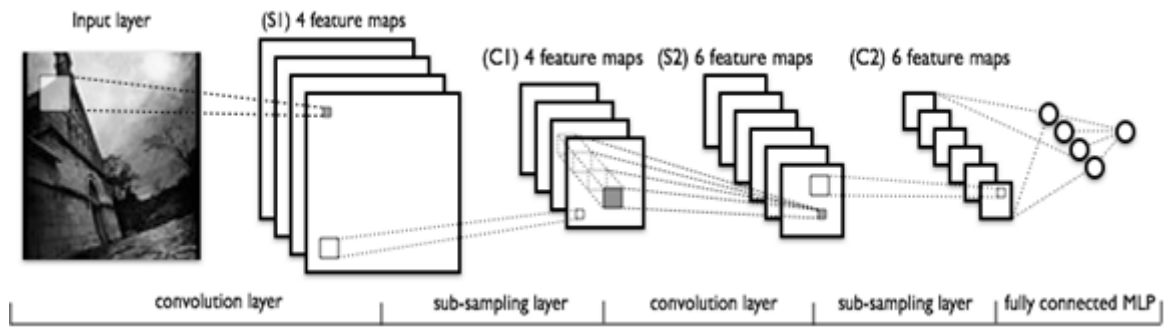


Рисунок 1.31 - Топологія згорткової нейронної мережі

Давайте розглянемо рисунок 1.31. Перші два типи шарів convolutional і subsampling, чергуються між собою і формують вхідний вектор ознак для нашого багатошарового персептрона.

Згорткові нейронні мережі являють собою деяку золоту середину між природними (біологічними) і звичайними багатошаровими нейронними мережами. На сьогоднішній день саме згорткові нейронні мережі дають найкращі результати в області розпізнавання зображень, в середньому точність розпізнавання даних нейронних мереж перевершує інші штучні нейронні мережі на 10 - 15% що дуже хороший результат.

В основі успіху згорткових нейронних мереж лежить концепція загальних ваг. Не дивлячись на великий розмір, ці мережі мають невелику кількість змінних, параметрів, що настраюються в порівнянні з іншими нейронними мережами і своїм предком - неокогнітроном. Також існують варіанти згорткових нейронних мереж, схожі на неокогнітрон, і ґрунтуються на зворотному поширенні помилки. Дані нейронні мережі також здатні до швидкого навчання.

На рисунку 1.32 показано модель нейрона згорткової нейронної мережі. Дана модель допоможе нам на етапі вибору топології нейронної мережі.

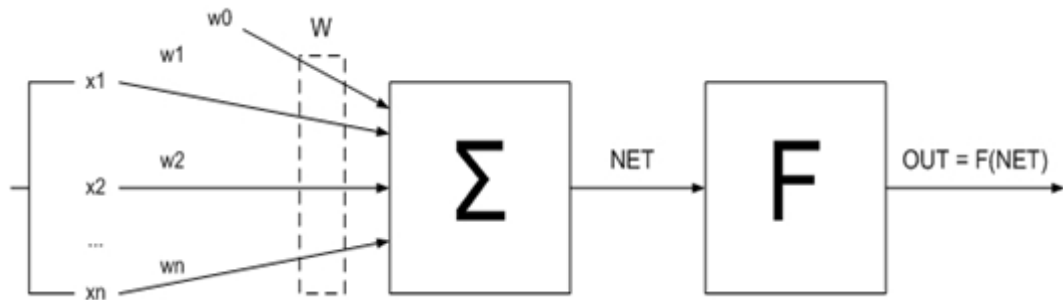


Рисунок 1.32 - модель нейрона згорткової нейронної мережі

Сам по собі вибір топології орієнтується в першу чергу на поставлене завдання, дані інших дослідників і особистий досвід. Можна виділити кілька етапів впливають на вибір топології:

- визначити поставлене завдання для нейронної мережі (прогнозування, модифікація, класифікація, кластеризація);
- визначити обмеження в розв'язуваній задачі (швидкість, точність відповіді);
- визначити вхідні дані.

Тепер настав час більш детально розглянути шари нейронної мережі. Для початку розглянемо вхідний шар. Вхідні дані являють собою зображення, знімки із супутників або аерофотозйомки. Всі зображення відібрані і мають цілий ряд спільних рис, зокрема - загальний план. Вхідний шар враховує двовимірну топологію зображень і складається з декількох карт або матриць, карта може бути одна в тому випадку, якщо зображення представлено в відтінках сірого, в інших випадках їх три, де кожна карта відповідає одному з кольорних (рисунок 1.33) каналів (червоний, синій, зелений) .

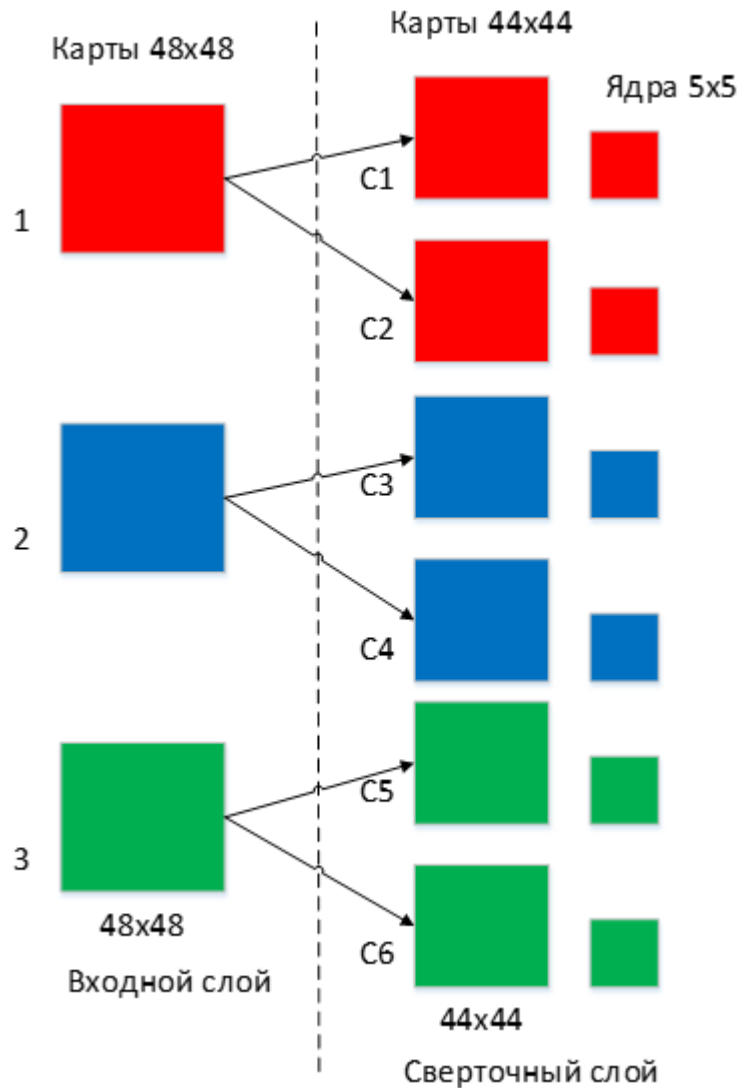


Рисунок 1.33 - Зв'язок згорткового шару з попередніми

Давайте тепер розглянемо згортковий шар. Сам по собі згортковий шар являє собою набір різного роду карт (їх ще іноді називають картами ознак, хоча по суті це звичайні матриці), у кожній з цих карт (матриці) є в наявності синаптичне ядро, або як його ще називають фільтром або скануючим ядром. Кількість карт визначається вимогами до задачі, якщо взяти велику кількість карт, то підвищиться якість розпізнавання об'єктів на зображеннях, котрі були використані як вхідні дані, однак зростуть вимоги до обчислювальної потужності, що може бути неприйнятним, і на протязі тривалого часу стримувало

повсюдне застосування нейронних мереж. Виходячи з наявних на даний момент наукових статей, найбільш вдалим рішенням буде брати співвідношення один до двох, отже кожна матриця попереднього шару пов'язана з двома картами згортками у відповідності до рисунку 1.33.

На рисунку 1.33 ядро являє собою фільтр або вікно, яке ковзає по всій області попередньої карти і знаходить якісь ознаки об'єктів. Наприклад, якщо навчати модель на безлічі супутникових знімків, то одне з ядер, наприклад, буде в процесі навчання давати найбільший сигнал в області даху будинку, автомобіля або дороги. Якщо розмір ядра занадто малий, то воно не може виділити будь-які ознаки, таким чином необхідно правильно підібрати розмір ядра. Також розмір ядра вибирається таким чином, щоб розмір карт згорткового шару був парний, це дозволяє не втрачати інформацію при зменшенні розмірності в підвибірковому шарі (рисунок 1.34).

Приклад ядра з навченою ознакою

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0



Вхідне зображення

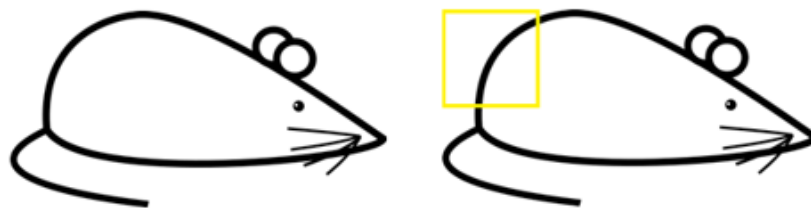


Рисунок 1.34 - Приклад навчання ядра

Ядро являє собою систему поділюваних ваг або синапсів, це одна з головних особливостей згорткової нейронної мережі. У згортчній незвичайній

мережі загальні ваги дозволяють скоротити число зв'язків і знаходити одну ознаку по всій області зображення в той час як у звичайній багатошарової мережі дуже багато зв'язків між нейронами.

Тепер розглянемо підвибірковий шар. Даний шар так само як і згортковий має карти, однак їх кількість співпадає з попереднім. Мета даного шару - зменшення розмірності карт попереднього шару. Якщо, наприклад, на попередній операції згортки були виявлені деякі ознаки то для подальшої обробки настільки докладне зображення вже не потрібно і його можна ущільнити до менш докладного (рисунок 1.35).

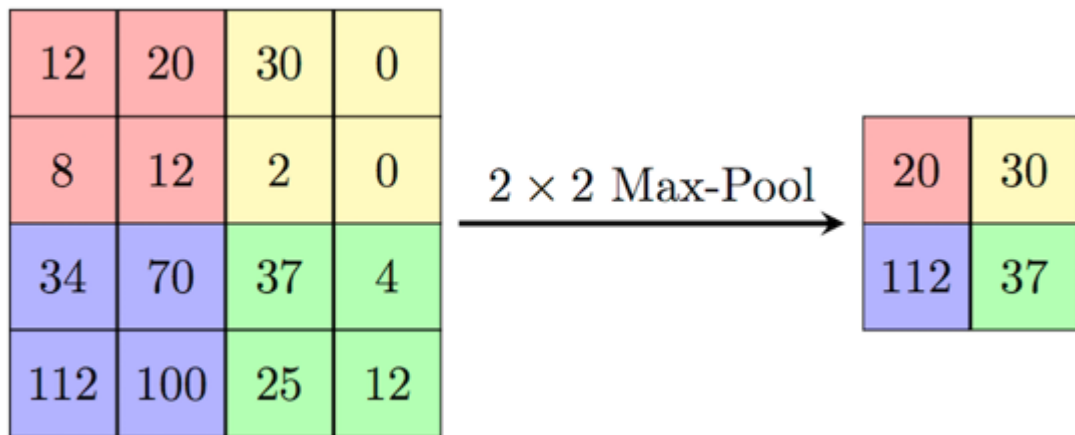


Рисунок 1.35 - Створення підвибіркового шару

Висновки до першого розділу

У даному розділі було розглянуто актуальність та сучасний стан картографії та її нового перспективного напрямку – автоматичної картографія, яка базується на алгоритмах розпізнання та сегментації зображення. Було приведено шляхи та технології отримання аерофотознімків. Розглянуто основні та

перспективні алгоритми сегментації зображення. Розглянуто принцип роботи штучних нейронних мереж, та задачі які можна вирішувати завдяки ним. В результаті було обрано будувати модель на базі нейронних мереж згорткової архітектури.

2 ОГЛЯД ДОСЛІДЖЕННЯ

2.1 Вхідні дані

Для проведення дослідження, було взято дані з змагальної платформи Kaggle, підготовленні Лабораторією оборонної науки і техніки (DSTL) Великої Британії, для проведення змагання «DSTL Satellite Imagery Feature Detection». В цьому змаганні кинуто виклик точно класифікувати об'єкти на аерофотознімках. Автоматична позначення об'єктів допоможе DSTL не тільки приймати швидкі та розумні рішення для захисту безпеки Великої Британії, а також привносити інноваційні методи комп'ютерного зору для аналізу супутникових зображень.

Сам по собі набір даних складається з 57 зображень, котрі було розділено на навчальну вибірку, котра складається з 25 зображень, та на тестову вибірку, котра складається з 32 зображень відповідно. Кожне з зображень, покриває приблизно один квадратний кілометр земної поверхні. Що стосується супутникових знімків, то знімки однієї і тієї ж території можуть бути різних видів та цінності з точки зору машинного навчання.

Об'єкти на знімках представленні у вигляді 10 класів:

- buildings (будівлі);
- structures (штучні приміщення, переважно паркани);
- road (асфальтовані автомобільні шляхи);
- track (грунтові шляхи);
- trees (дерева);
- crops (сільськогосподарські угіддя);
- waterway (річки);

- standing water (невеликі водойми та озера);
- vehicle large (вантажівки);
- vehicle small (легкові автомобілі).

Варто зауважити, що площа даних класів розподілено нерівномірно. На рисунку 2.1 представлена гістограма розподілення часток площ класів відносно загальної площі. Для наочності, на рисунку 2.2 представлено звичайний супутниковий знімок із набору даних, а на рисунку 2.3 представлене розмічене зображення того ж самого знімку.

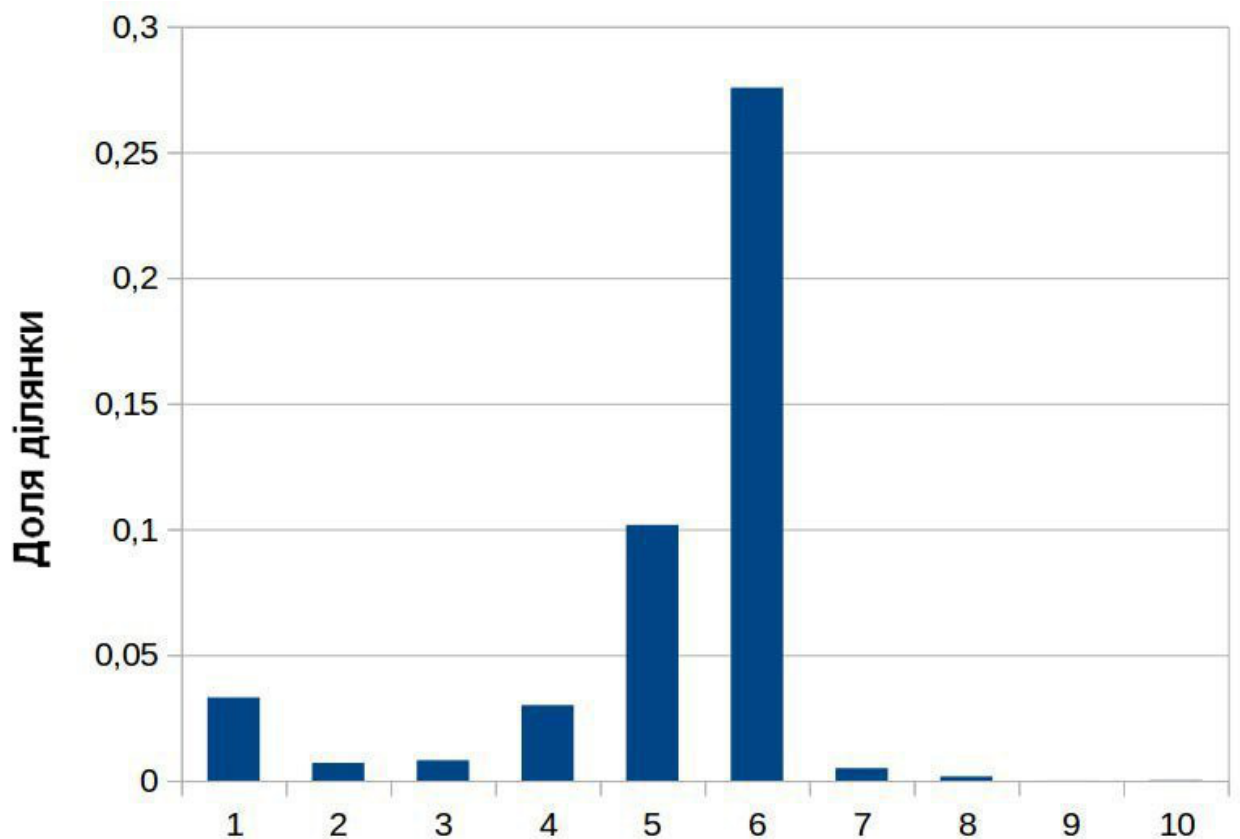


Рисунок 2.1 – Гістограма розподілення класів. 1 – building, 2 – structures, 3- road, 4 – track, 5 – trees, 6 – crops, 7 – waterway, 8 – standing water, 9 – vehicle large, 10 – vehicle small.



Рисунок 2.2 – Приклад знімку з набору даних

Варто зазначити, що вказані вище класи, як правило розподілені вкрай нерівномірно. Так, наприклад, деякі класи зустрічаються надзвичайно часто, в той же час інші - вкрай винятково (один піксель котрий відповідає автомобілю приходить на 60000 пікселів котрі визначені як посівні площі). Також, часто трапляється, що пікселі одного класу можуть бути перетинатися з пікселями іншого класу, як наприклад автомобіль на дорозі.

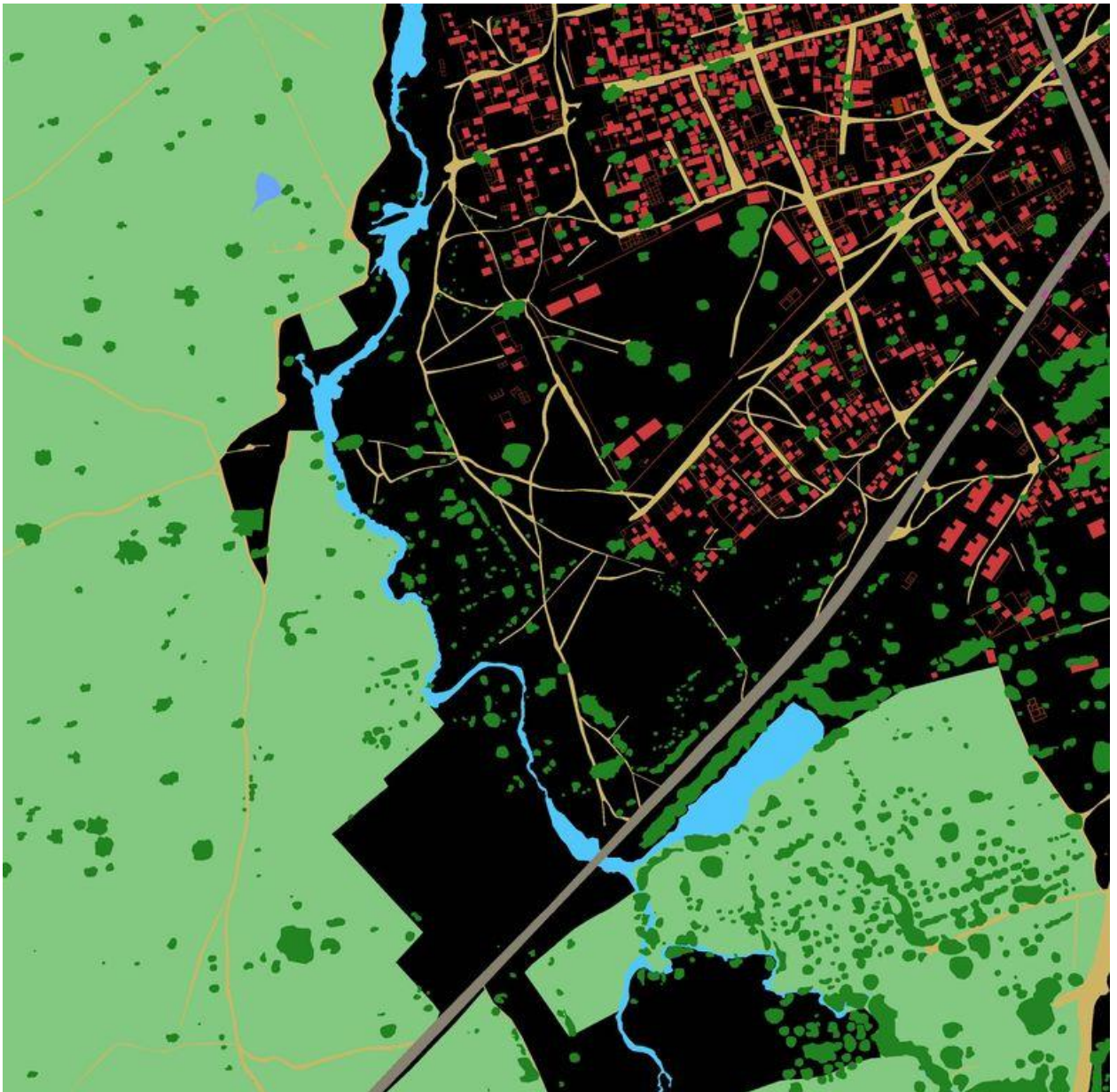


Рисунок 2.3 – Приклад розміченого зображення з тренувального набору даних. Червоний – building, помаранчевий – structures, сірий- road, жовтий – track, темно-зелений – trees, світло-зелений – crops, голубий – waterway, блакитний – standing water, фіолетовий – vehicle large, рожевий – vehicle small.

Кожне з зображень із набору даних представлено двадцятьма знімками відзнятими чотрма сенсорами супутника WorldView3 компанії DigitalGlob, а саме RGB-сенсор, панхроматичний сенсор, мультиспектральний сенсор, інфрачервоний сенсор SWIR. RGB та панхроматичні сенсори відповідно дають

звичайні кольорові та чорно-білі зображення. Мультиспектральні зображення надають можливість виокремлювати важливі деталі, котрі досить часто недосяжні для людського ока. Так, наприклад, інфрачервоний спектр зазвичай використовують для того, щоб виділяти різного роду рослинні об'єкти та їхні параметри. Подібне можливо, завдяки сильній рефлексії електромагнітного спектру рослинних об'єктів.

Наступні мультиспектральні діапазони можуть бути використані для розпізнавання певних класів об'єктів [1]:

- Coastal (400-452 nm) даний діапазон підходить для пошуку глибоких синіх та фіолетових “відтінків”, даний діапазон підходить для розпізнавання води а також димових та пилових хмар;
- Blue (448-510 nm) даний діапазон використовується для розпізнавання “нормального” синього відтінку, він підходить для кращого розпізнавання води а також відокремлення ґрунтових та кам'янистих поверхонь;
- Green (518-586 nm) цей діапазон використовується для розпізнавання зеленого “відтінку”, часто використовується для розмежування посівних площ та дерев;
- Yellow (590-640 nm) даний діапазон підходить для розмежування зелених насаджень та ґрунту, однак не може розділити воду та ліси, адже вони представляються одним тоном;
- NIR (772-954 nm) такий діапазон розпізнає ближній інфрачервоний спектр, даний діапазон особливо важливий, для різного роду екологічних досліджень;
- SWIR (1195-2365 nm) цей діапазон покриває різні ділянки короткохвильового інфрачервоного випромінювання, що використовується при розпізнаванні ступеню вологості поверхні, а також у геології, адже

мінерали в даному спектрі легко відрізнити одне від одного;

Далі в таблиці 2.1 буде представлена детальна інформація стосовно супутникових знімків представлених у вхідному наборі даних.

Таблиця 2.1 – Характеристика зображень супутника World View 3 представлених у вхідному наборі даних.

Назва	Довжина хвилі, нм	Роздільна здатність, м	біт/піксель	Роздільна здатність файлів, піксель
Panchromatic	450-800	0.31	11	~ 3396x3348
RGB (Red)	630-690			
RGB (Blue)	450-510			
RGB (Green)	510-580			
Coastal	400-450	1.24		~ 849x837
Blue	450-510			
Green	510-580			
Yellow	585-625			
Red	630-690			
RedEdge	705-745			
Near-IR1	770-895			
Near-IR2	860-1040			
SWIR-1	1195-1225	7.5	14	~136x134
SWIR-2	1550-1590			
SWIR-3	1640-1680			
SWIR-4	1710-1750			
SWIR-5	2145-2185			
SWIR-6	2185-2225			
SWIR-7	2235-2285			
SWIR-8	2295-2365			

2.2 Архітектура нейронної мережі U-net

Після проведення всебічних досліджень в області пошуку найбільш ефективних і достатніх засобів для досягнення поставлених цілей. Таким рішенням для завдання сегментації зображень було обрано нейронну мережу архітектури U-net. Це рішення було прийнято на підставі цілої низки спостережень і результатів серед інших дослідників. На рисунку 2.4 представлено результати змагання конкурсу ISBI 2015, який полягає у вирішенні задачі сегментації клітин на зображеннях з оптичного мікроскопу [9]. Як видно, дана нейронна мережа істотно перевершує майже всіх своїх конкурентів за кількістю безпомилкових проходів свого завдання, а саме завдання розпізнавання об'єктів на різного роду зображеннях шляхом проведення сегментації зазначених вище зображень.

Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	0.9203	0.7756

Рисунок 2.4 - Порівняння U-net з іншими підходами у задачі трекінгу біологічних клітин у змаганні ISBI 2015

U-net вважається однією із стандартних архітектур згорткових нейронних мереж для задачі сегментації зображення, коли потрібно не тільки визначити клас зображення цілком, но і потрібно сегментувати його область по

класу, тобто створити маску, яка буде розділяти зображення на декілька класів. Структурна схема архітектури нейронної мережі U-net представлена на зображенні 2.5. Кожен синій квадрат відповідає багатоканальній карті властивостей. Кількість каналів приведено у верхній частині квадрата. Розмір x-y наведено в нижньому лівому краю квадрата. Білі квадрати представляють собою копії карти властивостей. Стрілки позначають різні операції.

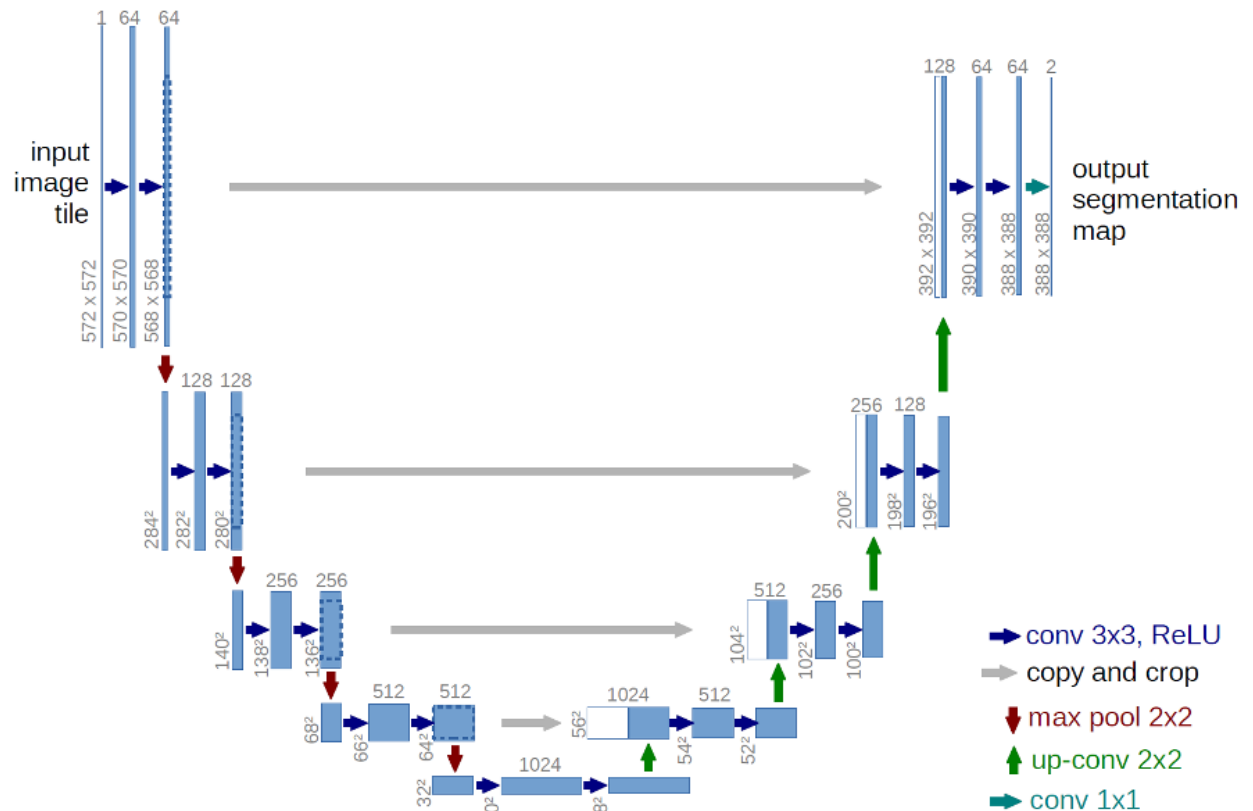


Рисунок 2.5 - Архітектура мережі U-net

Для нейронної мережі U-net характерно досягнення високих результатів в різних задачах та використання відносно невеликого об'єму даних для досягнення належних результатів.

Архітектура мережі являє собою послідовність шарів згортки та субдискретизації, які спочатку зменшують просторову роздільну здатність картинки, а потім збільшують його, попередньо об'єднавши з даними картинки і пропу-

стивши через інші шари згортки. Таким чином, мережа виконує роль своєрідного фільтра. Конкретні ознаки, які добуваються тією або іншою картою ознак, визначаються в процесі навчання нейронної мережі з учителем [10].

На кожному етапі понижувальної дискретизації канали властивостей подвоюються. Кожен крок у розширенні шляху складається з операції підвищує дискретизації карти властивостей, за якою слідує:

- 1) згортка 2×2 , яка зменшує кількість каналів властивостей;
- 2) об'єднання з відповідним чином обрізаної картою властивостей з стягують шляху;
- 3) дві 3×3 згортки, за якими слідує ReLU

На останньому шарі використовується згортка 1×1 для зіставлення кожного 64-компонентного вектора властивостей з бажаним кількістю класів. Всього мережа містить 23 згорткових шарів.

2.2.1 ReLU

Rectified linear unit або випрямляч, у контексті глибоко навчання, є функцією активації яка визначається наступним чином (2.1):

$$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} = x^+ = \max(0, x) \quad (2.1)$$

Функція є кусочно-лінійна. Перш за все зазначимо, що ReLU-нейрони ефективніше заснованих на логістичному сигмоїді і гіперболічному тангенсі. Наприклад, щоб підрахувати похідну, потрібно обчислити непросту функцію а потім ще виконати операцію множення, з тангенсом приблизно та ж історія,

тільки потрібно зводити в квадрат. А щоб обчислити похідну $\text{ReLU}'(x)$, потрібно рівно одне порівняння: якщо x менше нуля, видаємо нуль, якщо більше нуля, - одиницю.

На перший погляд це здається несуттєвим, але на практиці означає, що засновані на ReLU-нейронах мережі при одному і тому ж «обчислювальному бюджеті» на навчання, на одному і тому ж «залізі» можуть бути значно більше (за розміром, тобто по числу нейронів), ніж мережі з більш складними функціями активації. Однак сам по собі цей аргумент мало що значить: врешті-решт, щоб підрахувати похідну ступінчастої функції, навіть порівнювати ні з чим не треба. Треба, щоб нова конструкція працювала і дійсно навчалася.

2.2.2 Згортковий шар

Згортковий шар реалізує ідею локальних рецептивних полів, кожен вихідний нейрон з'єднаний тільки з певною (невеликий) областю вхідний матриці і таким чином моделює деякі особливості людського зору [11]. В спрощеному вигляді згортковий шар можна записати наступною формулою (2.2):

$$x_j^l = f\left(\sum_i x_i^{l-1} * k_j^l + b_j^l\right) \quad (2.2)$$

де

x_j^l – карта ознак j (вихід слою l),

$f()$ – функція активації,

b_j – коефіцієнт здвику для карти ознак j ,

k_j – ядро згортки номер j ,

x_i^{l-1} – карта ознак попереднього шару.

При цьому, за рахунок крайових ефектів розмір вихідної матриці зменшується. Вихідний сигнал j, k -ого нейрона, в рамках однієї карти ознак обчислюється за формулою (2.3) так званого ядра або фільтра згортки:

$$\sigma(b + \sum_l \sum_h w_{l,h} a_j + l, k + h) \quad (2.3)$$

Одна карта ознак зі своїм ядром ознак розпізнає лише одну ознаку. Для розпізнавання іншого признаку потрібно додати ще одну карту ознак зі своїм власним ядром. На практиці, внутрішні шари включають в себе десятки та сотні карт ознак.

Початкове зображення подається на вхідний шар. У першому шарі підвибірки кожна карта ознак здійснює пошук певної, закріпленої тільки за даною карткою, ознаки. Досягається це за рахунок використання загальної для всієї карти ознак матриці ваг і особливої організації локального рецептивного поля для кожного нейрона такої карти.

Кожен нейрон карти ознак отримує вхідні дані від прямокутної області розміру $n \times m$ вхідного зображення. Така область досить мала і безліч таких областей на вхідному зображенні перетинаються і накладаються за принципом черепиці.

Суміжні нейрони карти ознак отримують в якості вхідного впливу суміжної прямокутної області, причому вагові коефіцієнти для всіх нейронів карти ознак будуть однаковими. Для простоти викладу будемо називати область, котра формує локальне рецептивне поле нейрона шару підвибірки, вікном. Відповідно, площу вікна вважати - кількістю нейронів в такій області.

Таким чином, можна говорити про те, що карта ознак в цілому здійснює операцію пошуку ознаки у вхідних даних. Інші карти ознак мають

інший набір вагових коефіцієнтів i , відповідно, здійснюють пошук інших ознак у вхідних даних.

Після виконання згортки мережа втрачає частину інформації про точне положення знайденої ознаки, але зберігає інформацію щодо взаємного розташування різних ознак.

2.2.3 Субдискретизуючий шар

Даний шар виконує задачу зменшення розміру вхідної карти ознак (звичай у два рази). Це можливо робити різними способами, в даному випадку, розглянемо метод вибору максимального елемента (max-pooling). Формально вся карта ознак поділяється на вікна розміром 2×2 елемента, з яких обираються максимальні за значенням. Формально, даний шар може бути описаний за наступною формулою (2.4):

$$x^l = f(a^l \times \text{subsample}(x^{l-1}) + b^l) \quad (2.4)$$

де

x^l – вихід шару l ,

$f()$ – функція активації,

a, b – коефіцієнти,

subsample – операція вибірки локальних максимальних значень.

Використання даного шару дозволяє покращити розпізнавання образів зі зменшеним масштабом (зменшеним чи збільшеним) [8]. Як правило шар субдискретизації чергуються зі гортковими шарами.

2.3 Метрики оцінювання

Для оцінювання результатів використовувалась метрика коефіцієнт Жаккара (Jaccard Index), котрий розраховується за наступною формулою (2.5) [12]:

$$J = \frac{TP}{TP + FP + FN} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (2.5)$$

де

TP – true positive,

FP – false positive,

FN – false negative.

Схематично формулу (2.5) можливо зобразити за допомогою кіл Ейлера, що представлено на рисунку 2.6.

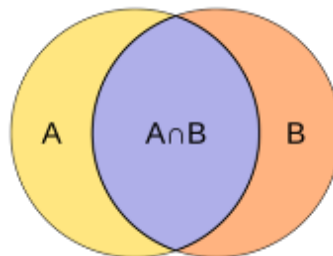


Рисунок 2.6 – Перетин реального полігону А, з прогнозованим полігоном В

З формули (2.1), очевидно що коефіцієнт Жаккара змінюється на проміжку від 0 до 1. В даному дослідженні, метрика рахувалась одночасно для всіх перевірочних завдань. Тому погане прогнозування на одному зображенні мало вплив на фінальний показник метрики. Тому бажано уникати помилково-позитивних спрацювань.

2.4 Навчання нейронної мережі U-net

На вхід нейронної мережі подаються двадцять каналів зображення розміром 144 на 144 пікселів. Нейронна мережа навчається методом стохастичного градієнтного спуску на основі вхідних зображень та відповідних карт сегментації. Вихідне зображення має дещо менші розміри (на постійну ширину границі) ніж вхідне, за рахунок згортки. Далі, функція soft-max попіксельно розраховує енергію по остаточній карті властивостей, разом з функцією крос-ентропії. Крос-Ентропія в кожній точці розраховується за наступною формою (2.6):

$$E = \sum_{x \in \Omega} \omega(x) \log(p_{l(x)}(x)) \quad (2.6)$$

Границя розділення розраховується з використанням морфологічних операцій. Потім розраховуються карта вагових коефіцієнтів за наступною формулою (2.7):

$$\omega(x) = \omega_c(x) + \omega_0 \times \exp\left(-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2}\right) \quad (2.7)$$

де

ω_c - карта вагів для балансування частот класів,

d_1 - відстань до границі найближчої соти,

d_2 - відстань до границі другої найближчої соти.

Для того, щоб провести ітерацію навчання, необхідно зрозуміти як через них виражається значення градієнтів функції помилки ваг. Через функцію

отримання максимуму помилка проходить без змін, шар дискретизації не виконує навчання. Однак, прохід через граф розраховані градієнти розрідженими, адже з усіх елементів вікна субдискретизації $z_{i,j}^l$, приватна похідна $\frac{\partial E}{\partial x_{i,j}^{l+1}}$ відноситься тільки до одного, максимального, а всі інші отримають нульовий градієнт, на цьому процес навчання можна вважати завершеним.

Далі, необхідно виконати процедуру пропускання через не лінійність:

$$\frac{\partial E}{\partial y_{i,j}^l} = \frac{\partial E}{\partial z_{i,j}^l} \frac{\partial z_{i,j}^l}{\partial y_{i,j}^l} = \frac{\partial E}{\partial z_{i,j}^l} h'(y_{i,j}^l) \quad (2.8)$$

На даному етапі відомо $\frac{\partial E}{\partial z_{i,j}^l}$, а $h'(y_{i,j}^l)$, можна отримати шляхом розрахунків. В свою чергу, на згортковому рівні з'являються ваги, котрі потрібно буде навчати. Певна складність тут полягає у тому, що всі ваги діляться і кожен приймає участь у всіх результатах, таким чином отримана сума досить значна (2.9):

$$\frac{\partial E}{\partial w_{a,b}^l} = \sum_i \sum_j \frac{\partial E}{\partial y_{i,j}^l} \frac{\partial y_{i,j}^l}{\partial w_{a,b}^l} = \sum_i \sum_j \frac{\partial E}{\partial z_{i+a,j+b}^{l-1}} \quad (2.9)$$

Де i та j проходять через всі елементи зображення на проміжному шарі $y_{i,j}^l$, тобто після згортки, але до субдискретизації.

Для повноти розуміння залишилось лише пропустити градієнти через попередній шар (2.10):

$$\frac{\partial E}{\partial x_{i,j}^l} = \sum_a \sum_b \frac{\partial E}{\partial y_{i-a,j-a}^l} \frac{\partial y_{i-a,j-a}^l}{\partial x_{i,j}^l} = \sum_i \sum_j \frac{\partial E}{\partial y_{i-a,j-b}^l} w_{a,b} \quad (2.10)$$

Варто звернути увагу, що зворотній прохід для згортки виявився дуже схожим на згортку з тими ж вагами $w_{a,b}$, тільки замість $i + a$ та $j + b$ тепер $i -$

a та $j - b$. В тому випадку, коли зображення доповнюються нулями, розмірності зберігаються, зворотній обхід можна вважати такою ж згорткою, як і прямий прохід, тільки розвернутими осями.

Для навчання нейронної мережі представлено всього 25 розмічених зображень. Після фрагментації даних зображень до розміру 144×144 пікселів отримаємо приблизно 10 тисяч фрагментів, котрі не перетинаються, це все одно дуже мало. В наслідок цього, було використано методику розширення навчальної множини (data augmentation). Так як, процес обертання супутникового знімку є інваріантним, тобто після повороту супутникового знімку на довільний кут, отримаємо коректне супутникове зображення. У якості аугментації було використане обертання на довільний кут, вертикальне та горизонтальне віддзеркалювання, що в решті решт допомогло збільшити вхідний набір даних у 8 разів, приблизно до 82 тисяч двадцяти каналних зображень 144×144 пікселів.

З метою запобігання ефекту перенавчання (overfitting), було використано метод перехресної перевірки (cross-validation) та метод виключення (dropout).

Для успішного навчання аналітичної моделі необхідно, щоб класи було представлено в тренувальній множині, в однакових пропорціях. Однак, якщо даних недостатньо, чи процедура семплінгу не дала очікуваних результатів, один з класів може виявитись домінуючим. Це може призвести до перекосу в процесі навчання і як наслідок домінуючий клас буде розглядатись як найбільш вірогідний. Метод перехресної перевірки дозволяє уникнути подібних ситуацій.

Перехресна перевірка має дві основні переваги перед підходом котрий полягає у використанні однієї множини для навчання та однієї множини для тестування моделі: розподілення класів стає більш рівномірним, що поліпшує

якість навчання. Якщо на кожній ітерації оцінити похибку моделі та узагальнити її по всіх ітераціях то отримана оцінка буде найбільш достовірною.

Суть методу виключення полягає в тому, що в процесі навчання обирається шар, з якого вилучається певна кількість випадкових нейронів (наприклад 30%). Такий підхід дозволяє покращити ефективність навчання та точність результатів. Найбільш натреновані нейрони отримують більшу вагу. По такій розрідженій мережі проводиться навчання, для залишених ваг робиться градієнтний спуск шаг, після чого всі виключені нейрони повертаються на свої місця. Таким чином, на кожному кроці стохастичного градієнта налаштовується одна з можливих 2^N архітектури мережі, де під архітектурою розуміється структуру зв'язку між нейронами, а через N позначається сумарна кількість нейронів.

2.5 Огляд інструментів

Для проведення подібних досліджень потрібні істотні обчислювальні ресурси. В даному випадку досліди проводилися на орендованій машині з 64 гігабайтами оперативної пам'яті і 11 гігабайтами відео пам'яті на науковій відеокарті Nvidia K80 Tesla GPU.

Для створення кодової бази нейронної мережі використовувався фреймворк TensorFlow. Лістинг коду представлений у Додатку А.

2.6 Огляд отриманих результатів

Розглянемо візуальне представлення роботи нейронної мережі, котре зображено на рисунку 2.7. В даному, окремо взятому випадку розглядаються можливості нейронної мережі для розпізнавання будівель. Верхня пара зображень показує еталонний результат розпізнавання (будівлі виділено білим кольором), в той же час нижня пара зображень показує результати роботи нейронної мережі.

Дивлячись на результат розпізнавання будівель, можна з упевненістю стверджувати, що нейронна мережа впоралась з завданням, та розпізнала будівлі з високою точністю.

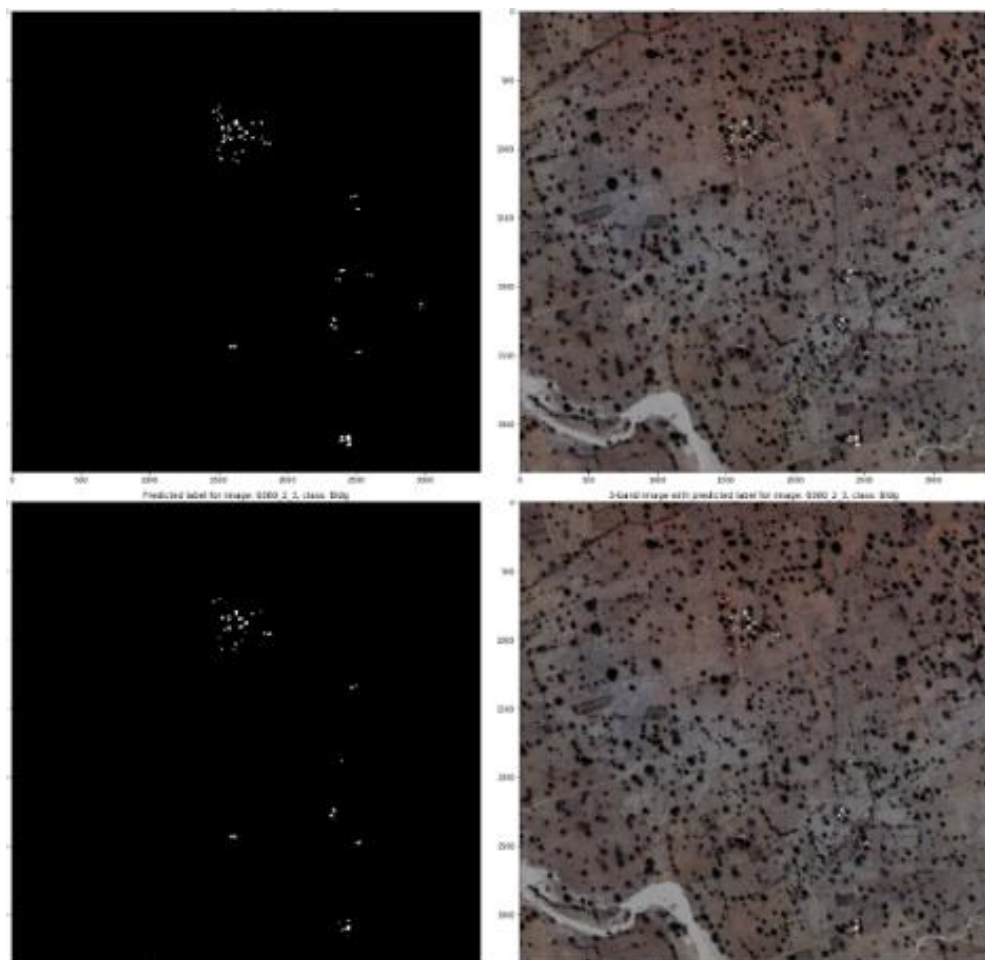


Рисунок 2.7 – Розпізнавання будівель

Далі, переходимо до розгляду роботи нейронної мережі в контексті розпізнавання структур (рисунок 2.8). Варто зазначити, що при розпізнаванні структур нейронна мережа працює на досить низькому рівні. В одних випадках вона не розпізнає такого роду об'єкти, в інших же розпізнає об'єкти, котрі належать до інших класів.

Вірогідно така ситуація зумовлена нашаруванням пікселів котрі належать до різних класів, а також низькою помітністю (контрастом) структур на знімках. Таким чином, можна дійти висновку, що поточна точність розпізнавання структур вимагає подальшого доопрацювання перед потенційним промисловим використанням.

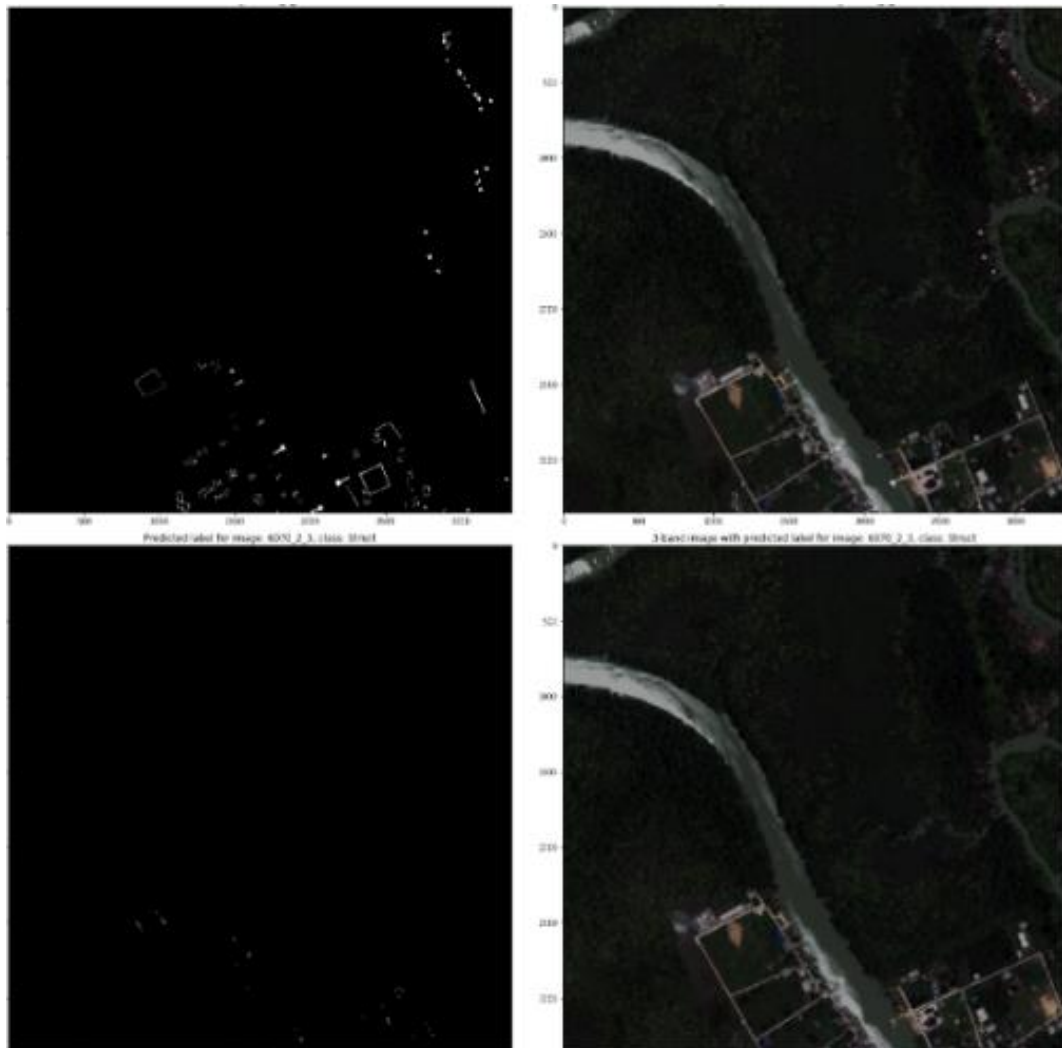


Рисунок 2.8 – Розпізнавання структур

Тепер, розглянемо процес розпізнавання автомобільних доріг (рисунок

2.9). На відміну від інших класів автомобільні дороги досить легко розпізнати, оскільки вони, як правило неперервні та яскраво виділяються на фоні зображення.

Тим не менш, навіть в такому сприятливому для розпізнавання класі можуть виникнути труднощі: дороги з активним транспортним трафіком можуть постраждати від перетину пікселів, котрі належать до різних класів. При правильному підході подібна властивість може поліпшити точність розпізнавання доріг.

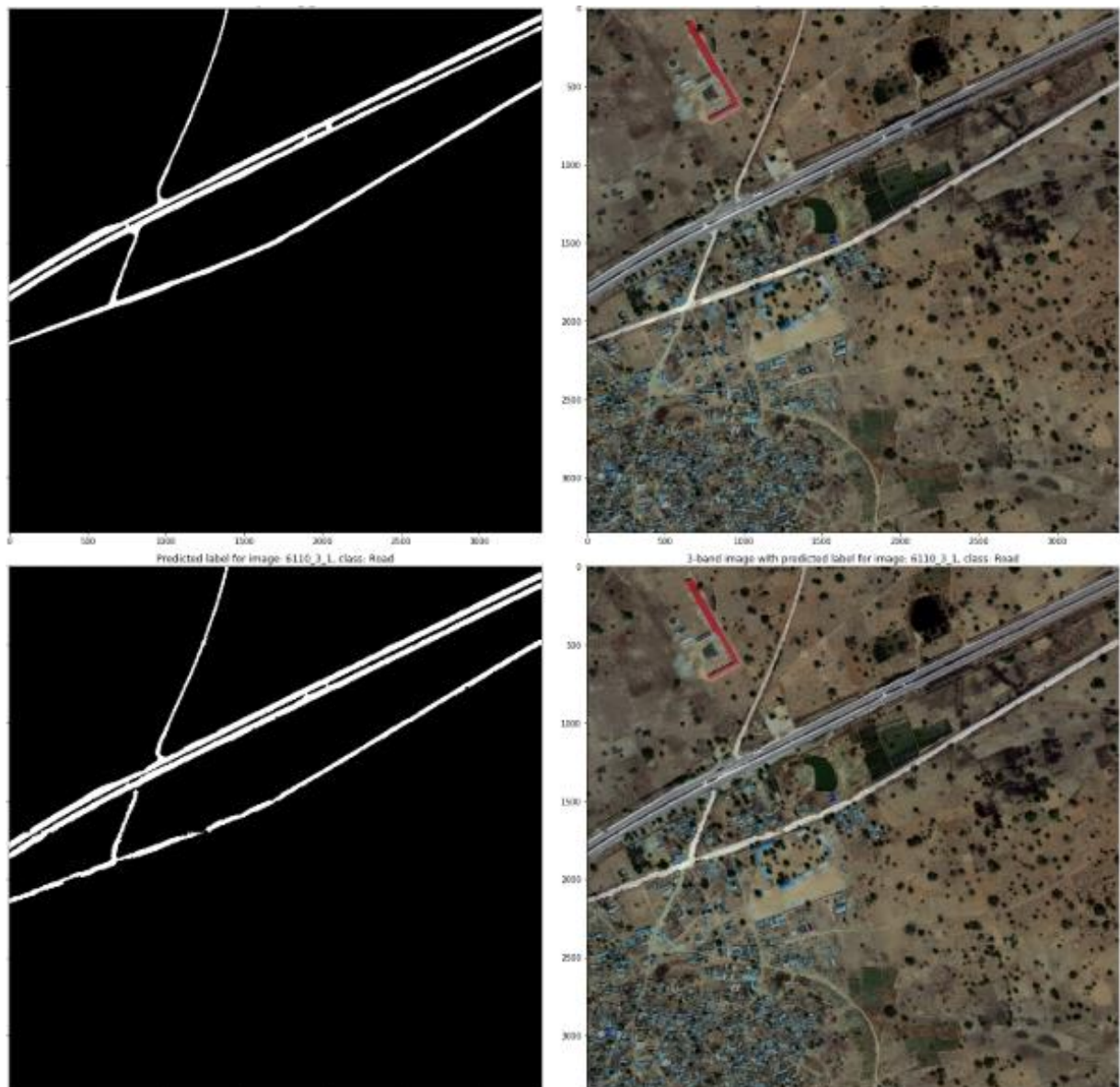


Рисунок 2.9 – Розпізнавання автомобільних доріг
На відміну від розглянутих вище автомобільних магістралей, інший клас

автомобільних доріг на багато менше виділяється на фоні місцевості. Рисунок 2.10 наочно демонструє складнощі котрі виникають при розпізнаванні ґрунтових доріг.

Ґрунтові дороги досить складно розпізнати не тільки через те, що вони слабо виділяються на місцевості, але й через те, що вони бувають досить хаотичними, адже як правило прокладаються місцевим населенням самостійно без проведення проектувальних робіт. Також дані дороги кардинально змінюють свій вигляд по мірі зміни сезонів. Так, наприклад, взимку ґрунтові дороги практично неможливо помітити.

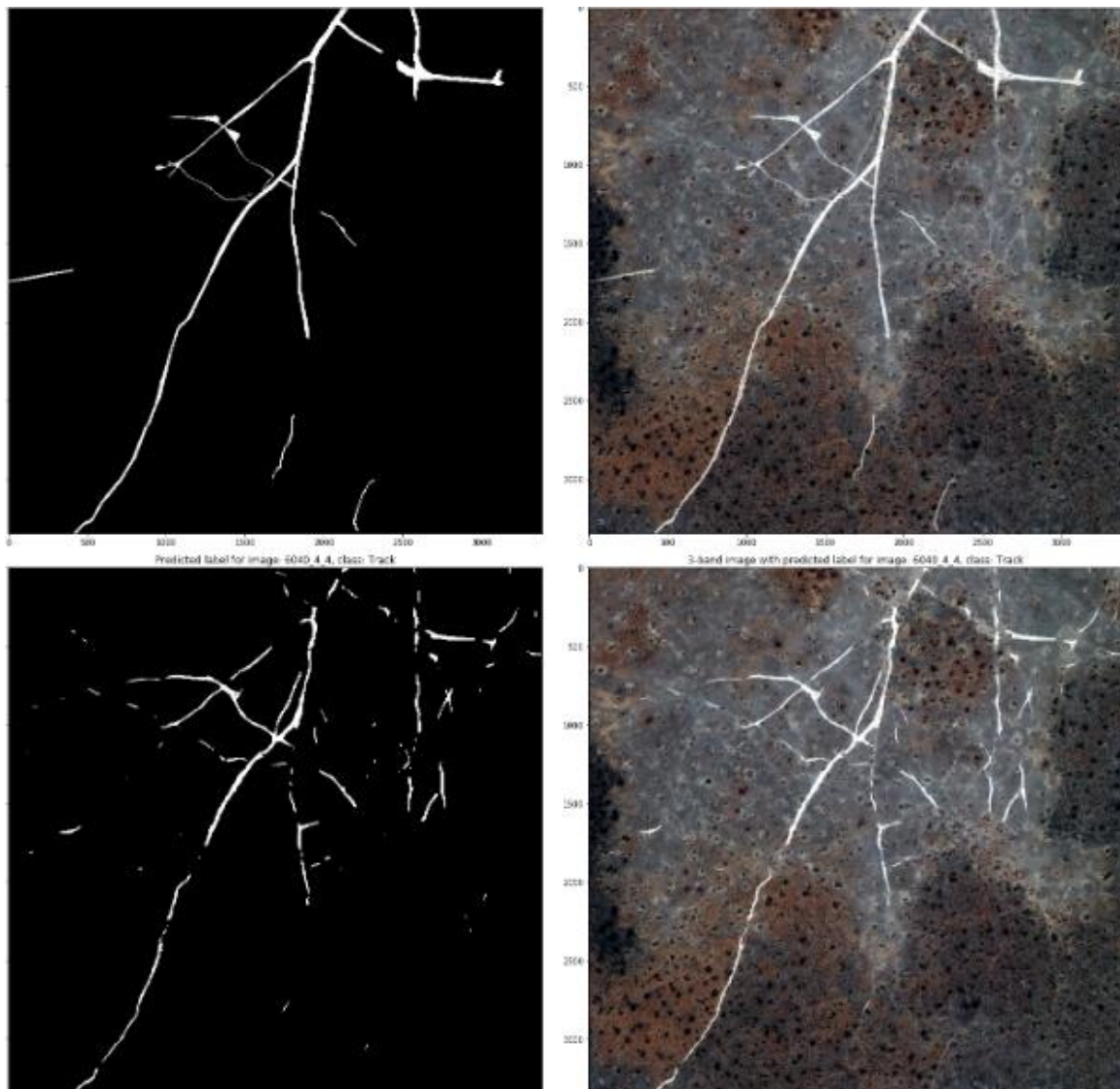


Рисунок 2.10 – Розпізнавання ґрунтових доріг
Настав час розглянути інший, дещо пов'язаний з ґрунтовими дорогами

клас - дерева. До даного класу відносяться як поодинокі дерева так і різного розміру зелені насадження від лісосмуг до густих лісів.

Варто зауважити, що даний клас досить легкий в розпізнаванні, головним чином за рахунок використання інфрачервоного каналу в супутникових знімках. Таким чином нейронна мережа з високою точністю розпізнає зелені насадження (рисунок 2.11).

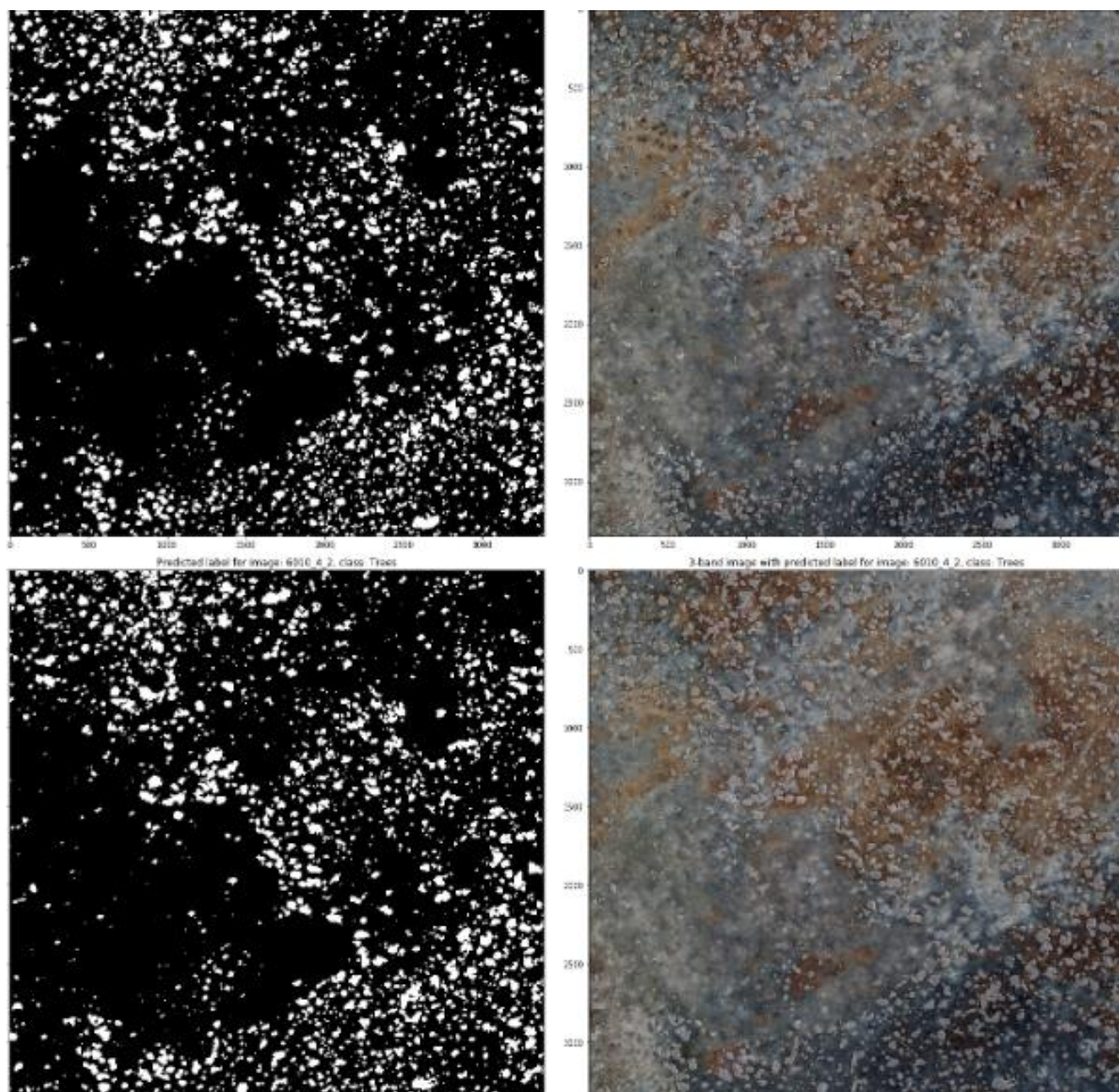


Рисунок 2.11 – Розпізнавання зелених насаджень

Останнім з представлених до розгляду класів є клас, котрий відповідає за розпізнавання посівних площ (рисунок 2.12). Даний клас не виділяється з точки зору ефективності, та знаходиться приблизно на одному рівні з іншими, хоча і поступається класу з розпізнавання автомобільних доріг.

Однією з особливостей даного класу є схильність до поглинання сусідніх класів, в першу чергу структур та ґрунтових доріг, зумовлено це неймовірною перевагою в розмірах з боку посівних площ та відносно малими розмірами структур та ґрунтових доріг. Також має місце і сезонний фактор.

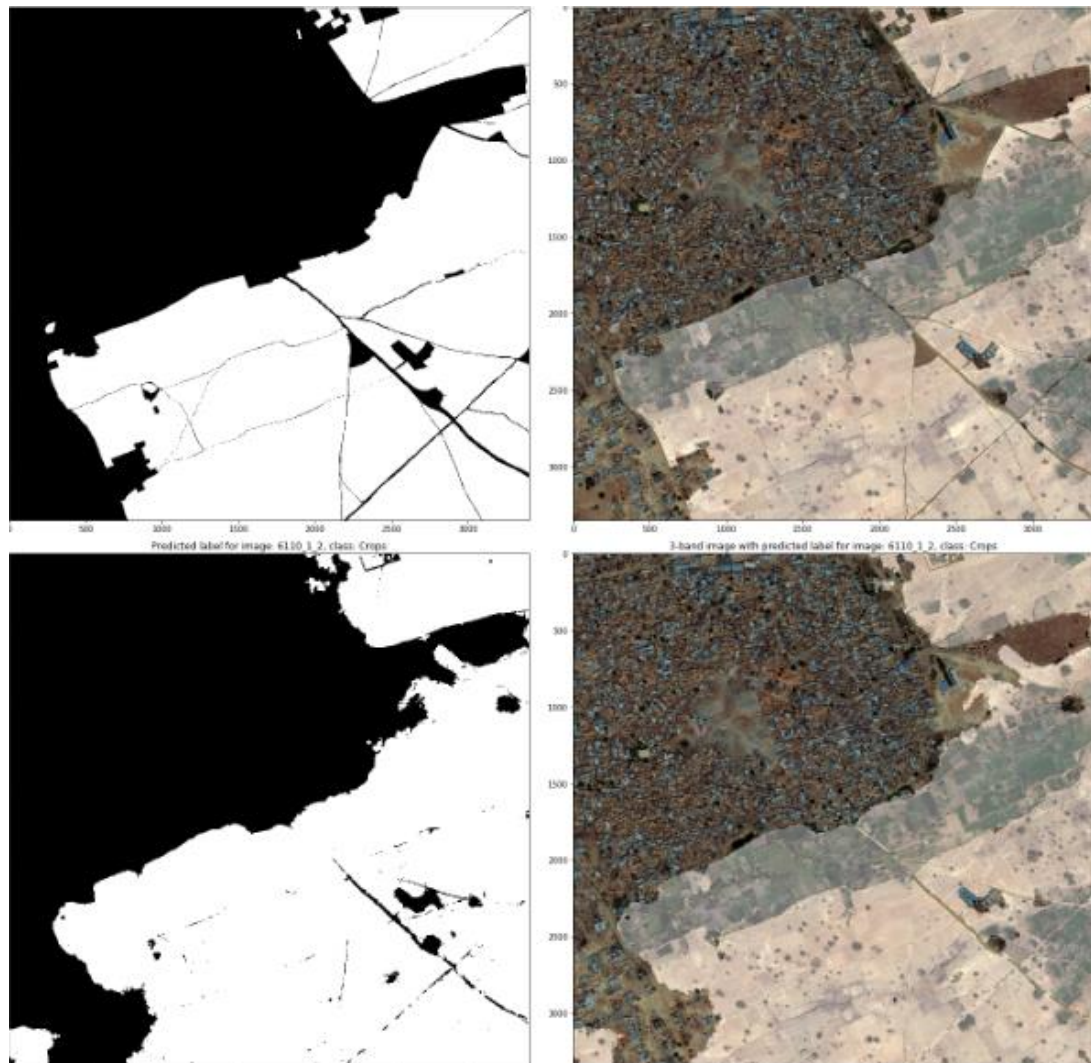


Рисунок 2.12 – Розпізнавання посівів

2.7 Підсумки проведеного дослідження

На таблиці 2.2 зображено результати роботи нейронної мережі в числовій формі, (коефіцієнт Жаккара). Загалом, можна стверджувати, що нейронна мережа успішно працює з більшістю класів, хоча і мають місце помилки в більш складних для розпізнавання, але й менш затребуваних класах.

Таблиця 2.2 – коефіцієнт Жаккара сегментації класів на валідаційному наборі даних

Клас	коефіцієнт Жаккара
Buildings	0,7832
Structures	0,2690
Road	0,7744
Track	0,6865
Trees	0,5931
Crops	0,8103
Waterway	0,5681
Standing water	0,4708
Vehicle large	0,1113
Vehicle small	0,0321
Загальний	0,4454

Загалом навчання даної нейронної мережі зайняло 10000 епох, та тривало 3432 хвилин, або приблизно 57 годин. Подібні результати, на момент проходження змагання, могли б потрапити в десятку фіналістів Kaggle. На таблиці 2.3 приведено значення оцінюваної метрики на навчальному та валідаційному наборах даних, кожні 1000 епох навчання нейронної мережі.

Таблиця 2.3 – значення коефіцієнту Жаккара на навчальному та валідаційному наборах даних

Кількість епох	Тренувальний набір даних, індекс Жаккара	Валідаційний набір даних, індекс Жаккара
1000	0,4434	0
2000	0,5427	0
3000	0,7435	0,1722
4000	0,6546	0,1364
5000	0,8548	0,2622
6000	0,8744	0,3325
7000	0,841	0,4063
8000	0,8943	0,4242
9000	0,9134	0,4353
10000	0,8862	0,4454

На рисунку 2.13 представлено графік динаміки навчання моделі на даних таблиці 2.3.

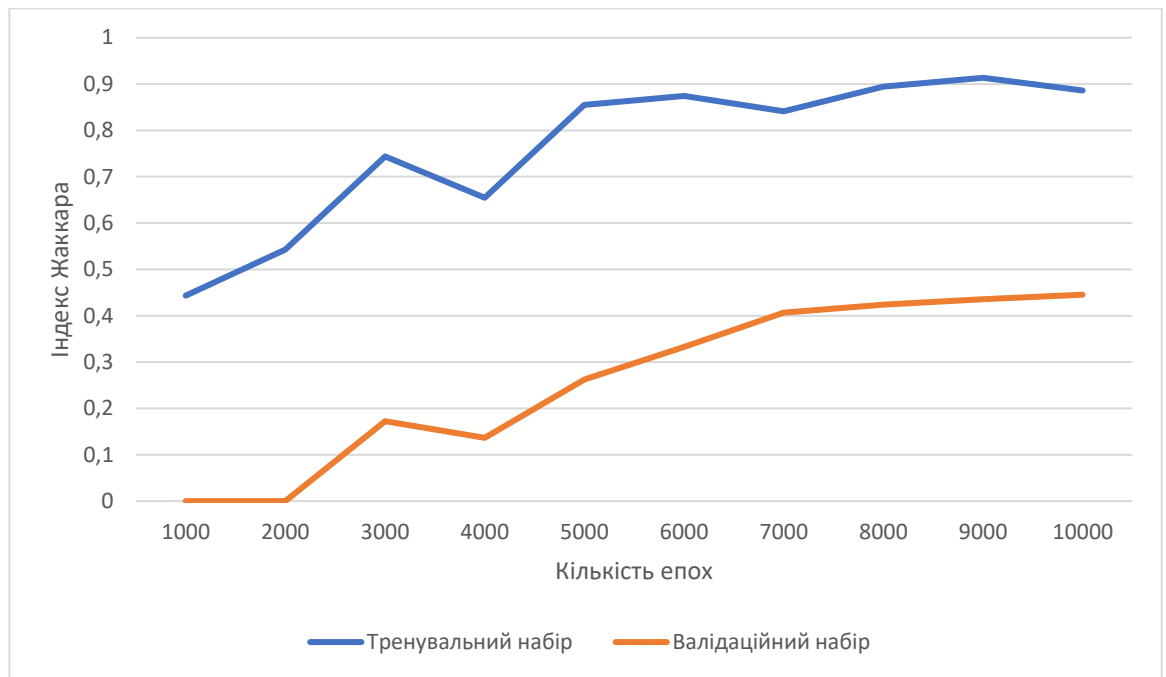


Рисунок 2.13 – Динаміка навчання моделі

Висновки до другого розділу

В даному розділі було розглянуто основні особливості нейронної мережі U-net, а саме компонентів з яких вона складається та алгоритм навчання.

Ключові моменти:

- у якості тренувального набору було обрано набір даних зі змагання «DSTL Satellite Image Feature Detection», який складається з 57 двадцяти каналних супутникових знімків;
- для дослідження обрано десять класів (buildings, structures, road, track, trees, crops, waterway, standing water, vehicle large, vehicle small);
- оптимальним вибором для виконання завдання виявилась нейронна мережа архітектури U-net;
- у якості міри оцінювання ефективності роботи нейронної мережі обрано

індекс Жаккара;

- середній показник натренованої моделі на валідаційному наборі даних за індексом Жаккара складає: 0.4454;
- в результаті випробувань наступні класи розпізнаються найкраще (за індексом Жаккара): crops - 0.81, buildings - 0.78, road - 0.77.

3 СТАРТАП

3.1 Передмова

Третій розділ даної роботи являє собою повний та послідовний аналіз перспектив, загроз та додаткових можливостей використання результатів наукової діяльності проведеної в рамках магістерської роботи, у якості стартап-проекту.

Сам проект ґрунтується на розроблений для розпізнавання різного роду об'єктів на аерофотознімках, шляхом використання згорткових нейронних мереж.

В ході маркетингового дослідження було проведено цілу низку різносторонніх досліджень, у тому числі прискіпливий аналіз пошук та аналіз потенційних ринків збуту, окреслено та досліджено коло потенційних споживачів, проведено пошук та аналіз конкуруючих та суміжних проектів, знайдено можливі конкурентні переваги над схожими проектами, виділено їхні слабкі та сильні сторони.

Так, наприклад було виділено три основні гравці ринку, а саме: Planet - сервіс, котрий займається цілодобовим моніторингом земної поверхні, Orbital insight - займає схожу нішу збираючи та аналізуючи дані супутникових знімків, ДеерМар - сервіс значною мірою схожий на попередній, однак він концентрується на автодорогах та підтримує можливість 3d візуалізації.

Далі, в даній роботі буде освітлено хід та результати маркетингового дослідження, в контексті розвитку дипломного проекту у якості стартапу.

3.2 Пошук та аналіз ідей

Дослідження, котре лежить в дипломного проекту може бути використано у багатьох напрямках, навіть у тих, котрі слабо пов'язані з дослідженням супутникових знімків. Однак, доцільно виділити основні, та найбільш комерційно привабливі вектори застосування, а саме: моніторинг господарської діяльності, екстрена картографія, військова картографія.

З огляду на долю аграрної складової в економіці нашої держави, а також наявність лісових та інших, доступних для розпізнавання засобами аерофотозйомки, ресурсів впливає потреба в контролі за їх використанням. Результати проведеного дослідження в купі з перспективами співробітництва з виробниками та постачальниками безпілотних літальних засобів дають змогу вирішити ці проблеми. Так, наприклад спеціалізований дрон, обладнаний відеокамерою достатньої якості, матиме змогу оцінювати стан полів, та повідомляти у разі виявлення ушкоджень нанесених крадіями, негодою чи шкідниками. Також, крім моніторингових перспектив, система розпізнавання в купі з системою позиціонування, встановлена на відповідний літальний апарат зможе розпиляти різного роду добрива чи отруту для шкідників, в автономному чи напів-автономному режимі. Окрім того, подібний підхід дозволить уникнути помилок при обробці угідь, а саме уникнути випадкового обприскування чужих чи не підготовлених посівів.

До списку пріоритетних напрямків розвитку було включено екстрену картографію. Суть даного напрямку полягає у тому, щоб при першій необхідності, в автоматичному режимі створити карти місцевості, постраждалої від стихії. Так, наприклад, можна буде відстежити пошкоджені буревієм дороги, лісові та степові пожежі, котрі часто мають місце влітку, на півдні нашої держави.

Окрім мирних, цивільних напрямків застосувань також варто виділити військовий, особливо з огляду останніх подій на теренах нашої держави. Модель, отримана в результаті проведення дослідження в даній дипломній роботі може розпізнавати велику кількість об'єктів, в тому числі і військових, а саме: крупну бойову техніку, вантажівки, окопи, дзоти. Таким чином дана модель, при відповідній модифікації може бути використана при створенні автоматичних дронів для моніторингу ворожих позицій у зоні бойових дій.

Ідеї стартапу, напрямки застосування та потенційні вигоди для користувачів зображено в табличному вигляді далі, на таблиці 3.1.

Таблиця 3.1 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Автоматизований аналіз аерофотознімків, та даних отриманих зі супутника.	Моніторинг господарської діяльності	Отримання актуальної інформації стосовно стану посівів, лісових ресурсів тощо.
	Екстрена картографія	Динамічне картографування постраждалих в результаті природних чи техногенних катастроф регіонів
	Військова картографія, розпізнавання техніки	Динамічне нанесення на карту різного роду об'єктів, котрі знаходяться в зоні бойових дій, класифікація даних об'єктів. Так, наприклад можливе розпізнавання крупної військової техніки: танки, артилерія тощо.

Також, дане рішення може бути інтегровано до програмних засобів безпілотних літальних апаратів у якості допоміжного інструменту, що дозволить зменшити вірогідність людської похибки при проведенні авіарозвідки.

Всі напрямки застосування в деякій мірі подібні між собою. При цьому цільові користувачі даних напрямків суттєво відрізняються між собою: так,

наприклад екстрена картографія цікавить представників міністерства надзвичайних ситуацій, а моніторинг господарської діяльності має дещо ширше коло потенційних споживачів, в їх числі фермери, екологічні служби, логістичні компанії.

Всі розглянуті вище ідеї, незважаючи на певні розбіжності, при своїй роботі користуються одним і тим же принципом - розпізнаванням об'єктів на аерофотознімках. Отже, всі найбільш доцільним вибором буде використання так званої стратегії “дробової рушниці”, тобто розвивати ідею як сервіс загального призначення, котрий здатний адаптуватись під будь-яку суміжну задачу у стисненій проміжок часу та з мінімальними витратами. Так, наприклад, маючи досвід навчання моделі для розпізнавання пробок на дорогах, можна в короткий строк створити модель здатну розпізнавати окопи та військову техніку. Що стосується розпізнавання різного роду катастроф, та створення екстрених карт, то тут також модель може бути перенавчена та налаштована в досить стислі строки. Дана особливість моделі дозволяє швидко реагувати на нові можливості та виклики.

Ключовим чинником конкурентної боротьби в даному сегменті є баланс технологічної переваги та розумної цінової політики. Хоча, в деяких випадках саме технічні питання, такі як точність та швидкодія суттєво перевищують властивість цінової складової. Тим не менше, не зважаючи ні на що, завжди потрібно шукати золоту середину, адже будь який проект повинен бути гармонічно розвиненим.

На таблиці 3.2 розглянуто дану ідею, та її переваги над потенційними конкурентами.

Таблиця 3.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
	Мій проект	Planet	Orbital insight	DeepMap			
Швидкість розпізнання	Швидке розпізнавання простих класів	Швидко працює, кількість класів обмежена	Середні витрати часу, стандартна візуалізація	великі витрати часу, однак висока якість візуалізації			+
Візуалізація	Фарбування відповідних об'єктів	Фарбування відповідних об'єктів	Виділення об'єктів на карті	3D візуалізація деяких об'єктів		+	
Гнучкість	Сервіс загального призначення, можливе донавчання моделей	Сервіс орієнтований на моніторинг	Сервіс загального призначення	Сервіс орієнтований на дороги			+
Здатність до інтеграції	Легка інтеграція даного сервісу	Інтеграція можлива шляхом використання API	Інтеграція можлива шляхом використання API	Інтеграція можлива шляхом використання API			+

Також, що стосується технологічного аспекту неодмінно варто підкреслити важливість інтеграційних можливостей та перспектив програмного продукту, або його складових частин, котрі тим чи іншим чином можуть бути використані при розробці різного роду стороннього програмного забезпечення різного характеру. Так, наприклад, навчена нейронна модель може на пряму використовуватись на стадії програмування базового функціоналу безпілот-

них літальних апаратів, а саме записуватись до пам'яті пристрою з метою забезпечення подальшого автоматичного розпізнавання об'єктів в автономному режимі, тобто без підключення до всесвітньої мережі інтернет. Подібного роду безпілотні літальні апарати, окрім застосування у цивільному секторі можуть застосовуватись і в військових операціях у якості дронів-розвідників, або атакуючих дронів-камікадзе. Однак, у випадку військового використання дана модель потребуватиме певного роду доопрацювання, а саме, вона повинна бути навчена розпізнавати дружні цілі, та уникати їхнього ураження, що враховуючи підхід до маркування техніки не складно, оскільки розпізнавальні знаки досить великого розміру та чітко на несені.

Далі, в таблиці 3.3, буде розглянуто технологічний аспект. Як впливає з приведених вище даних, саме технологічна складова є основною гарантією успішності проекту, головним чином, з тої причини, що даний проект орієнтований на вузький прошарок суспільства, для якого стандартні засоби маркетингу не ефективні, і головну роль грають технічні характеристики продукту.

Таблиця 3.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технологія її реалізації	Наявність технології	Доступність технології
1	Розпізнавання різного роду об'єктів та явищ на супутникових знімках поверхні землі	Мануальна розмітка	+	+
2		Прямі алгоритми комп'ютерного бачення	+	+
3		Штучні нейронні мережі	+	+

Як було зазначено в таблиці вище, всі необхідні технології для реалізації

даного проекту нам доступні. Таким чином даний набір технологій дозволяє правильним чином навчити модель, котра буде ефективно виконувати покладені на неї завдання, а саме розпізнавати об'єкти на супутникових знімках, використовуючи при цьому необхідний мінімум розрахункових та часових ресурсів.

3.3 Аналіз ринку

Цільовою аудиторією даного стартапу є представники державних органів (служба надзвичайних ситуацій, екологічна служба, збройні сили прикордонна служба тощо.) а також представники середнього та великого бізнесу, чия діяльність прямо або опосередковано пов'язана з крупними площами, такими як посівні площі сільськогосподарських культур, лісозаготівля, картографія тощо.

Однак, на жаль, на даний момент визначити ємність ринку не є можливим, однак, є можливість оцінити вартість контракту, шляхом оцінки вимог а також вартості необхідної інфраструктури, такої як дрони придатні до аерофотозйомки або супутники, здатні робити знімки земної поверхні достатньої якості.

Таблиця 3.4 показує основні характеристики, та вимоги груп користувачів. Варто зазначити що між двома групами потенційних споживачів суттєва різниця в інтересах, так наприклад, у випадку з міністерством надзвичайних ситуацій, для розпізнавання лісової пожежі достатньо не надто дорогих та високоякісних засобів аерофотозйомки, котрі будуть робити знімки низької якості, адже контраст між палаючим та цілим лісом досить значна, та не призведе проблем з сегментацією.

Таблиця 3.4 - Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Моніторинг господарських угідь	Фермери, лісозаготівники, контролюючі служби	Поведінка даних груп клієнтів схожа, різниця полягає у рівні деталізації.	Доступний інтерфейс, підтримка широкого спектру аерофотографічних апаратів
2	Авіаційна обробка посівів, садів, лісів	Фермерські, садові, лісові господарства	Поведінка всіх груп, в даному випадку досить схожа, так наприклад, авіаційна обробка посівів носить сезонний характер у всіх випадках	Розпізнавання типу, належності угідь. Обробка відповідних угідь (вимога не до програмного забезпечення, а до самого літального апарату)
3	Екстрена картографія	Міністерство надзвичайних ситуацій	Можливі довгострокові, але не надто дорогі контракти	Оперативність, надійність
4	Військова картографія, розвідка	Міністерство оборони	Вірогідна закупівля безпілотних літальних апаратів	Оперативність, надійність

Тим не менше, бізнес зацікавлений у набагато більш точному, детальному та якісному аналізу, та готовий платити за якість, хоча і надає перевагу саме оренді подібних засобів, ніж покупці коштовного обладнання. Дана форма зацікавленості пов'язана, головним чином, з сезонністю сільськогосподарської діяльності та бажанням розвантажити власні адміністративні ресурси, уникнути необхідності створення необхідної ремонтної та експлуатаційної інфраструктури, котра досить вимоглива в обслуговуванні та не перетинається з традиційним сільськогосподарським обладнанням. Хоча, можливі винятки, наприклад покращення (модифікація) вже існуючих та придбаних користувачем апаратів (оприскувальних безпілотних та пілотованих апаратів).

Загалом, всі вимоги прийнятні, і незважаючи на різницю між користувачем, система достатньо гнучка для адаптації.

На даному етапі потрібно виділити основні фактори ризику для даного проекту. Актуальні фактори ризику буде розглянуто нижче, на таблиці 3.5. Тим не менше, варто виділити найсерйозніші з них, а саме перший та другий.

Перший пункт, нестача або дорожнеча засобів аерофотозйомки. У випадку нестачі обладнання неможливо вести активну господарську діяльність, а саме - отримувати своєчасні знімки місцевості. Відсутність знімків, робить систему для їх розпізнання недієздатною. У випадку дорожнечі засобів аерофотозйомки буде знижено попит і на наше програмне забезпечення.

Другим фактором загрози є проблема пов'язана безпосередньо з роботою самої моделі, а саме низька якість розпізнання. Тобто, дана модель на практиці може виявитись не ефективною, та не надавати користувачеві точної та вичерпної інформації. В такому випадку комерційна привабливість проекту суттєво зменшується.

Ці фактори ризику (таблиця 3.5) суттєво пов'язані між собою, адже точність розпізнавання часто залежить не тільки від нетренованої моделі, але й від якості даних котрі отримані на вхід з метою подальшого аналізу, в свою ж

чергу якість отриманих зображень на пряму залежить від якості обладнання за допомогою якого дані знімки було отримано.

Таблиця 3.5 - Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Нестача(до- рожнеча) за- собів для ае- рофотозйо- мки	Через нестачу засобів аеро- фотозйомки можливі ускладнення з постійним моніторингом	Пошук альтернативних мо- жливостей оренди апарату- ри, співпраці з виробни- ками апаратури тощо.
2	Неточне роз- пізнання	При великій кількості схо- жих між собою випадків (входу на розпізнання) мо- жливі проблеми з корект- ним розпізнаванням	Виділення та навчання спе- ціалізованих моделей, пок- ращення документації (по- сібників користувача)
3	Активна кон- куренція	Можлива активізація кон- курентів.	Пошук шляхів підвищити привабливість сервісу, по- кращити інтерфейс, просу- вання на виставках, зма- ганнях
4	Скорочення державного фінансування	Відсутність у цільових мі- ністерств коштів на закупі- влю програмного забезпе- чення (техніки, де дане програмне забезпечення використовується у якості основного чи допоміж- ного)	Диференціювання ринків збуту заздалегідь: робота з іноземними державними органами, сконцентрувати зусилля на завоюванні у першу чергу клієнтів з приватного сектору ринку
5	Спад попиту на продукцію аграрного се- ктору	Відсутність коштів на роз- виток у гравців аграрного бізнесу внаслідок скоро- чення попиту на їхній то- вар	Диференціювання ринків збуту: пошук нових мож- ливостей застосування та кооперації, налагодження співпраці з клієнтами котрі належать до державного сектору економіки.

В деяких випадках значення мають і навички оператора, у випадку з пілотованими літальними засобами чи дистанційно керованими безпілотними літальними апаратами.

Активна конкуренція, в свою чергу є найменш серйозною з представлених вище загроз, однак, легке подолання даного фактору можливе лише в тому випадку, якщо два попередніх не задіяні, або задіяні слабо.

Загалом, стартап цілком здатен впоратись з кожним з можливих факторів ризику.

Наступним кроком, слід розглянути різного роду фактори можливостей даного проекту. На відміну від вказаних вище факторів загроз, фактори можливостей являють собою можливість отримання певних конкурентних переваг, та потенційний розвиток успіху.

Всі вказані на таблиці 3.6 фактори так чи інакше пов'язані з співпрацею з виробниками та постачальниками самого обладнання для аерофотозйомки чи для обробки сільськогосподарських угідь, або ж з підприємствами котрі надають відповідні послуги. Також більша частина можливостей залежить від наявності та важливості видобування чи захисту певних природних ресурсів, так, наприклад, пожежа у лісі котрий містить цінні породи дерева, котрі є об'єктом промислу, обробки та експорту, може завдати суттєвої шкоди не тільки екологічній ситуації в країні, але й вдарити по економічній складовій, що робить моніторинг подібних ресурсів пріоритетним, і як наслідок збільшує попит на послуги даного стартап-проекту, та фінансування подібного роду послуг. З наведених вище причини, одним з ключових факторів успішності стартап-проекту є детальний та повний аналіз потенційних покупців, аналіз економіки, природних умов на території потенційних клієнтів. Також поруч з завданням раннього виявлення лісових пожеж лежить задача моніторингу потенційних незаконних вирубок цінних лісових порід, таким чином, при вказаних вище умовах попит зростає ще більше.

Таблиця 3.6 - Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Висока активність з боку природних катастроф	Через складну погодні ситуацію може зрости попит на автоматизовані засоби моніторингу (наприклад лісові пожежі, влітку)	Створення спеціалізованих рішень, пропозиції представникам відповідних відомств, створення запасів на випадок попиту.
2	Наявність великої кількості цінних лісових ресурсів	При наявності великої кількості лісових ресурсів виникає потреба в точному контролі та обліку лісозаготівлі.	Співпраця з контролюючими органами та зацікавленими громадськими організаціями
3	Збільшення попиту на ринку сільськогосподарських культур, а саме тих, котрі, вимагають посиленої турботи	Корисність даного стартап-проекту збільшується в залежності від стійкості та складності у догляді сільськогосподарських культур, наявності та впливові різного роду шкідників, тим самим підвищуючи потребу в автоматизованих засобах моніторингу та обробки (обприскування), де наявні технології можуть бути використані у якості модуля, відповідального за розпізнавання стану та приналежності посівів.	Активна кооперація з виробниками засобів для повітряної обробки посівів (пілотованих та не пілотованих літальних засобів а також, можливо, хімічних засобів), цільовий маркетинг.

Загалом, даний стартап проект має достатньо факторів можливості, щоб триматись на плаву та мати змогу розвиватись, а за умови мудрої політики стосовно вказаних вище факторів цілком можливо зайняти позицію лідера на ринку.

В цілому, можна сказати, що на внутрішньому ринку конкурувати досить легко, з огляду на пріоритет вітчизняного виробника. Однак, проблеми

можуть виникнути на глобальному ринку, тож для виходу на такий рівень потрібно буде докладно продумати маркетингову компанію, усунути недоліки та підкреслити переваги. Також, варто враховувати той факт, що даний стартап проект є інноваційним за своєю природою, а значить може бути сприйнятий з недовірою і єдиним способом нівелювання даного фактору є впровадження гнучкої цінової політики та використання короткострокових контрактів, котрі дають змогу клієнтам випробувати продукт без необхідності укладення довгострокових та дорогих контрактів. Тим не менше, в контексті даного стартапу впровадження подібної політики може бути досить складним з оглядом на питання пов'язані з обслуговуванням обладнання, що в черговий раз приводить нас до необхідності співпрацювати з поточними постачальниками обладнання з метою підвищити доступність продукту.

Виконуючи аналіз конкуренції в галузі за М.Портером варто розпочати з розгляду прямих конкурентів в галузі, а саме:

- Orbital Insight - сервіс з розпізнавання об'єктів, доріг та екстреної картографії загального призначення;
- DeepMap - сервіс з розпізнавання об'єктів на супутникових знімках загального призначення, однак з ухилом на автошляхи.

Окрім даних, прямих конкурентів існує один не прямий - Planet, даний сервіс головним чином надає доступ до супутникових знімків, однак можливості з розпізнавання не є профільними та досить обмежені.

Щодо постачальників, то ситуація складається досить позитивно, з огляду на велику кількість конкуруючих між собою постачальників, що призводить до появи вигідних з точки зору ціни та якості пропозицій.

Також, варто зауважити, що представники держави досить обмежені в можливостях розпоряджатись коштами а також здатні впливати на ринок шляхом формування правил ринкової діяльності. До категорії товарів-замінників

(таблиця 3.7) можна віднести проведення аналогічних дій шляхом використання традиційних людських ресурсів (пілотовані польоти, розгляд та обробка отриманих даних в ручному режимі).

Таблиця 3.7 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари заміники
	Orbital Insight, DeepMap	Planet	Виробники обладнання для аерофотозйомки	Середній і великий бізнес, міністерство надзвичайних ситуацій, контролюючі органи	Наглядачі з біноклями на гелікоптерах, прості дрони з камерами, пілотовані засоби авіаобробки
Висновки:	Конкуренція на внутрішньому ринку обіцяє бути менш напруженою ніж на глобальному, з огляду на походження конкурентів	Даний потенційний конкурент, при успішних переговорах може стати партнером а саме - постачальником знімків	Постачальників досить багато, тож умови на ринку досить комфортні	Клієнти, а саме державні установи можуть регулювати ринок шляхом законів, податків та зборів.	Можлива відсутність довіри до засобів машинного навчання, в наслідок чого клієнти можуть надавати перевагу живим людям

В свою чергу до клієнтів можна віднести такі крупні та середні установи, як приведені нижче:

- Міністерство надзвичайних ситуацій;

- середній агробізнес;
- великий агробізнес;
- Міністерство оборони.

Основними факторами конкурентоспроможності (таблиця 3.8) даного проекту є фактори технічні, а саме: швидкодія та гнучкість. Швидкодія свідчить про здатність даного проекту виконувати покладені на нього завдання на порядок швидше ніж конкуренти. Що стосується гнучкості, то система побудована таким чином, що з мінімальними витратами часу та розрахункових ресурсів можливо налаштувати модель (або створити нову аналогічну) для розпізнавання будь-яких об'єктів, або налаштувати для виконання суміжних завдань.

Таблиця 3.8 - Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Швидкодія	Модель машинного навчання представлена в даному проекті працює надзвичайно швидко
2	Гнучкість	Враховуючи архітектуру системи та нейронної мережі можна навчати різного роду моделі, адаптувати їх до різних завдань

На основі проведених досліджень можна виділити наступні сильні сторони нашого проекту(таблиця 3.9): швидкодія, гнучкість. Щодо слабких сторін, то основною з них є нестача та дорожнеча обладнання для аерофотозйомки, а також необхідність розробки якісного, зрозумілого інтерфейсу. Щодо потенційних можливостей варто особливо відзначити можливість співпраці з виробниками та орендодавцями обладнання для аерофотозйомки, просування продукту в державах, з нестабільною природною обстановкою, проблемами з

незаконною вирубкою лісу тощо. Щодо загроз, то головною з них є нестача, та обмежений доступ до обладнання аерофотозйомки, також існує загроза з боку розвинених конкурентів, особливо при конкуруванні на глобальному ринку.

Таблиця 3.9 - SWOT- аналіз стартап-проекту

Сильні сторони	Слабкі сторони
швидкодія, гнучкість	проблеми з доступністю інфраструктури аерофотозйомки
Можливості	Загрози
розширення на ринки держави зі специфічним кліматом, співпраця з виробниками обладнання для аерофотозйомки	активна конкуренція, недоступність обладнання для аерофотозйомки

3.4 Ринкова стратегія проекту

Насамперед варто зауважити, що з огляду на специфіку проекту загалом, а також на коло потенційних споживачів, доцільним є просування даного проекту саме шляхом презентацій на різного роду виставках, конкурсах, у рамках прямих переговорів з потенційними клієнтами. Зумовлений такий підхід на сам перед тим, що потенційні клієнти являють собою досить вузький суспільний прошарок, та не схильні прислухатись до інших форм маркетингу.

На даний момент очевидними є лише приведені нижче (таблиця 3.10) групи, оскільки зацікавленість інших потенційних клієнтів низька, або ж за-

надто низька їхня купівельна спроможність. Тим не менше, ситуація може змінитись по мірі здешевлення послуг аерофотозйомки. Так, наприклад, свого часу на самому початку розвитку автомобільного транспорту був досить низький попит на запасні частини для автомобілів.

Таблиця 3.10 - Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів прийняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Директори середніх та великих аграрних підприємств, або менеджер по закупівлям обладнання для них підприємств, високопоставлені агрономи тощо.	споживачі готові прийняти продукт, за умови високої точності	Попит залежить головним чином від зниження витрат на моніторинг стану посівів, за рахунок впровадження засобів автоматизованого моніторингу	Інтенсивність конкуренції низька, або на середньому рівні, з огляду на малу кількість конкурентів	середня
2	Представники міністерства надзвичайних ситуацій, відповідальні за попередження НС, та закупівлі. Аналогічного рангу представники організацій державного контролю	споживачі готові прийняти продукт, за умови надійності та дешевизни	попит залежить від частоти природних катастроф та від рівня контролю лісових ресурсів	Інтенсивність конкуренції низька, з огляду на низьку платіжездатність потенціальних клієнтів	Відносно проста
	Які цільові групи обрано: обидві цільові групи достойні уваги, однак друга група має пріоритет з огляду на простий вхід.				

Тим не менше, по мірі зростання доступності та розповсюдженості автомобільного транспорту створювалась різного роду інфраструктура, котра складалась зі станцій технічного обслуговування та АЗС. Те саме може чекати і на різного роду безпілотні літальні апарати, у тому числі і на ті, котрі безпосередньо можуть застосовуватися і у даному проекті.

На даний момент кількість альтернатив розвитку проекту невелика (таблиця 3.11), однак в майбутньому, з розвитком нейронних мереж, та збільшенню доступності обладнання стане можливим просування проекту на іншими шляхами розвитку.

Таблиця 3.11 - Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Середній та крупний агробізнес	диференційований маркетинг	швидкодія, гнучкість	стратегія диференціації
2	Державні моніторингові служби	концентрований маркетинг	низька вартість, надійність	стратегія лідерства по витратах

Враховуючи той факт, що даний проект безумовно є першопроходцем, щонайменше, на ринку України. Таким чином, оптимальною стратегією конкурентної поведінки буде так звана стратегія лідера. Що стосується пошуку споживачів, на внутрішньому ринку компанія зосередиться на приваблюванні саме перших споживачів, а не переманювання від інших, однак на міжнародному ринку доведеться також і боротись за клієнта. Скоріше за все компанія буде відслідковувати та запозичувати хороші практики у галузі користувацьких інтерфейсів.

3.5 Маркетингова концепція

Після проведення всіх необхідних маркетингових досліджень, та формування певної стратегії поведінки та окреслення цільових ринків потрібно перейти до дослідження проблем формування системи збуту.

В нашому випадку, скоріше за все, система збуту буде комбінованою (таблиця 3.12), оскільки даному проекту, окрім випуску програмного забезпечення доведеться активно співпрацювати з виробниками різного роду дронів, обладнання для аерофотозйомки, постачальниками послуг супутникової зйомки.

Таблиця 3.12 - Формування системи збуту

№	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Представники бізнесу більше схильні до використання послуг сервісу в оренду.	Гарантувати швидку та точну роботу програмного забезпечення	Велика	Надання доступу в оренду, на певний строк
2	Представники різного роду державних структур	Гарантувати надійність роботи системи	Низька, інколи - середня	Продаж готових одиниць виробу (програмного забезпечення та/або засобів аерофотозйомки)

Даний проект має досить широкий ринок збуту, котрий охоплює як приватних підприємців середнього та крупного масштабу так і різноманітні дер-

жавні органи. Подібний підхід дозволяє уникнути критичних втрат у разі скорочення того чи іншого ринку, а також зберегти проект на плаву за несприятливих умов, в той же час дозволяючи отримувати максимальні матеріальні та репутаційні дивіденди.

До недоліків можна віднести відносно вузьку направленість у приватному секторі, а також відсутність необхідної для ефективного функціонування інфраструктури, у тому числі нестачу кадрів на місцях.

Тим не менше, проект цілком життєздатний та має хороші ринкові перспективи та ринки збуту, головною запорукою успіху є активна співпраця з постачальниками необхідного для роботи системи обладнання.

Висновки до третього розділу

Стартап проект підготовлений на базі даної магістерської роботи має всі шанси на високу комерціалізацію, за рахунок мінімізації впливу негативних факторів, та максимізації позитивних. На даному етапі головним недоліком самого потенційного програмного продукту є досить складний інтерфейс, котрий вимагає доопрацювання. В той же самий час потенційний продукт має і яскраво виражені сильні сторони такі, як наприклад, висока точність розпізнавання та швидкодія.

Ринкова ситуація також виглядає досить сприятливою, виробництво обладнання продовжує вдосконалюватися, а на внутрішньому ринку практично відсутні конкуренти. Тим не менше, на глобальному ринку мають місце кілька сервісів, чия діяльність частково або суттєво схожа з функціоналом даного проекту, однак даний проект має свої переваги у порівнянні з конкурентами, а

саме: точність розпізнавання та швидкодія. Також, на ринку мають місце проекти-постачальники суміжних послуг, з якими можлива кооперація. Так, наприклад, кооперація можлива з сервісами супутникової зйомки.

Що стосується маркетингової концепції та просування продукту, то тут варто виділити той факт, що даний продукт орієнтується на відносно вузький прошарок суспільства, а саме на представників середнього та великого бізнесу, а також державних органів. Таким чином, маркетингова стратегія полягає у проведенні безпосередніх контактів зі споживачами, шляхом прийняття участі у різного роду тематичних виставках, конкурсах, змаганнях тощо.

ВИСНОВКИ ПО РОБОТІ

В ході виконання магістерської дисертації було проведення дослідження методів сегментації зображень, штучних нейронних мереж, сучасних методів аерофотозйомки та картографії.

Поставлені задачі:

1. Провести дослідження засобів та методів отримання супутникових знімків, їхніх характеристик та особливостей;
2. Дослідити існуючі засоби та підходи до сегментації та розпізнавання об'єктів на зображеннях;
3. Підібрати репрезентативну навчальну вибірку, котра дозволить провести навчання моделі з найвищою ефективністю;
4. Реалізувати алгоритм та модель, котрі придатні для автоматичного нанесення розмітки на супутникових знімках;
5. Провести дослідження моделі на основних класах об'єктів (будівлі, автомобільні шляхи, зелені насадження, структури, ґрунтові дороги, посіви).

Під час виконання поставлених задач проведено:

- дослідження засобів отримання супутникових знімків;
- аналіз супутникових знімків на предмет придатності для розпізнавання об'єктів;
- дослідження сучасних підходів до сегментації зображень та розпізнавання об'єктів;
- пошук оптимальної вибірки для навчання та випробування моделі;

- навчання обраної моделі на обраній вибірці, пошук шляхів оптимізації процесів;
- аналіз отриманої моделі як ядра для потенційного стартап-проєкту.

В результаті було отримано модель нейронної мережі, котра здатна розпізнавати встановлені класи об'єктів на супутникових знімках. Точність, отриманої мережі, за індексом Жаккара склала 0.4454 на валідаційному наборі даних по всім 10 класам. Деякі з класів, окремо, мають високу точність за індексом Жаккара, а саме клас «будівлі» - 0.7832, «автомобільні шляхи» - 0.7744, «сільськогосподарські угіддя» - 0.8103.

Рекомендації для послідовників:

- випробувати можливість комбінування декількох моделей нейронних мереж, комбінування моделей з алгоритмами сегментації тощо;
- розширити навчальну вибірку;
- дослідити можливість отримання високоякісних(тих, які не поступаються якістю супутниковим) зображень шляхом використання атмосферних безпілотних літальних апаратів;
- розробити зручний інтерфейс користувача.

ПЕРЕЛІК ПОСИЛАНЬ

1. Кашкин В. Б. / Дистанционное зондирование Земли из космоса. Цифровая обработка изображений. / Кашкин В. Б., Сухинин А.И. – М.: Логос, 2001. – 264 с.
2. Анисимов Б. В. / Распознавание и цифровая обработка изображений. / Анисимов Б. В., Курганов В. Д., Злобин В. К. – М: Высшая школа, 1983. – 149 с.
3. Яхьяева Г.Э. / Анализ и обработка изображений: принципы и алгоритмы. / Яхьяева Г.Э. – М.: Бином, 2006. – 316с.
4. Вежневек В. / Сегментация изображений. Курс «Введение в компьютерное зрение». / Вежневек В. – М: МГУ ВМК, 2006. – 312с.
5. Осовский С. / Нейронные сети для обработки информации / Осовский С. – М.: Финансы и статистика, 2002. – 344 с.
6. Круглов В. В. / Искусственные нейронные сети. Теория и практика. / Круглов В. В., Борисов В.В. – М.: Горячая линия - Телеком, 2002. – 382 с.
7. Object Detection with Deep Learning: A Review [Электронный ресурс] – Режим доступа: <https://arxiv.org/pdf/1807.05511.pdf>
8. Круглов В. В. / Гибридные нейронные сети. / Круглов В. В., Борисов В. В. – Смоленск: Русич, 2001. – 224 с.
9. U-Net: Convolutional Networks for Biomedical Image Segmentation [Электронный ресурс] – Режим доступа: <https://arxiv.org/pdf/1505.04597.pdf>
10. Fully Convolutional Networks for Semantic Segmentation [Электронный ресурс] – Режим доступа: <https://arxiv.org/pdf/1605.06211.pdf>
11. ImageNet Classification with Deep Convolutional Neural Networks [Электронный ресурс] – Режим доступа: <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
12. Обработка и анализ изображений в задачах машинного зрения. /

[Визильтер Ю.В., Желтов С.Ю., Бондаренко А.В. и др.] – М.: Физматкнига, 2010. – 672 с.

13. Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation [Электронный ресурс] – Режим доступа: <https://papers.nips.cc/paper/5858-decoupled-deep-neural-network-for-semi-supervised-semantic-segmentation.pdf>

14. Hypercolumns for Object Segmentation and Fine-grained Localization [Электронный ресурс] – Режим доступа: <http://home.bharathh.info/pubs/pdfs/BharathCVPR2015.pdf>

15. Very Deep Convolutional Networks For Large-Scale Image Recognition [Электронный ресурс] – Режим доступа: <https://arxiv.org/pdf/1409.1556.pdf>

16. Васильев В.И. / Интеллектуальные системы управления с использованием нейронных сетей. / Васильев В.И., Ильясов Б.Г. – Уфа: УГАТУ, 1997. – 92 с.

17. Гарбук С. В. / Космические системы дистанционного зондирования Земли. / Гарбук С. В., Гершензон В.Е. – М.: Изд - во А и Б, 1997. - 296 с.

18. Гонсалес Р. / Принципы распознавания образов. / Гонсалес Р. – М.: Мир, 1978. – 413 с.

19. Гонсалес Р. / Цифровая обработка изображений / Гонсалес Р., Вудс Р. - М.: Техносфера, 2005. – 1072 с.

20. ДеМерс Майкл Н. / Географические Информационные Системы. Основы. / ДеМерс Майкл Н. — М: Дата+, 1999. - 507 с.

21. Дьяконова С. В. / Анализ методов сегментации спутниковых изображений / Дьяконова С. В., Зайченко Ю.П. // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка – 2012. – No 57– С. 118 - 123.

22. Зайченко Ю. П. / Сегментация и обработка спутниковых изображений сверхвысокого разрешения. / Зайченко Ю. П., Дьяконова С.В. // Вісник ЧДТУ – 2012. – No1 – С. 8 - 12.

23. Зайченко Ю.П. / Нечеткие модели и методы в интеллектуальных

системах / Зайченко Ю.П. – К: Изд. Дом «Слово», 2008. – 354с.

24. Лопандя А.В. / Основы ГИС и цифрового картографирования. Учебно-методическое пособие. / Лопандя А.В., Немтинов В.А. – Тамбов: «Тамбовский государственный технический университет», 2007. – 72 с.

25. Макеева А.В. / Основы нечеткой логики. Учебное пособие для вузов. / Макеева А.В. – Новгород: ВГИПУ, 2009 – 59с.

26. Надеран С.В. / Сравнительный анализ методов сегментации спутниковых изображений. / Надеран С.В. // «Системный анализ и информационные технологии», SAIT – 2014. - С. 242.


```

        tf.contrib.layers.\
            variance_scaling_initializer()
    ) as sc:
        return sc

def build_pred(x_in, H, phase):
    ...

    This function builds the prediction model
    ...

    num_class = H['num_class']

    conv_kernel_1 = [1, 1]
    conv_kernel_3 = [3, 3]
    pool_kernel = [2, 2]
    pool_stride = 2

    early_feature = {}
    reuse = {'train': False, 'validate': True, 'test': False}[phase]

    with slim.arg_scope(argument_scope(H, phase)):
        scope_name = 'block_1'
        x_input = x_in
        num_outputs = 64
        with tf.variable_scope(scope_name, reuse=reuse):
            layer_1 = slim.conv2d(x_input, num_outputs, conv_kernel_3, scope='conv1')
            layer_2 = slim.conv2d(layer_1, num_outputs, conv_kernel_3, scope='conv2')
            early_feature[scope_name] = layer_2

        scope_name = 'block_2'
        x_input = slim.max_pool2d(layer_2)
        num_outputs = 128
        with tf.variable_scope(scope_name, reuse=reuse):
            layer_1 = slim.conv2d(x_input, num_outputs, conv_kernel_3, scope='conv1')
            layer_2 = slim.conv2d(layer_1, num_outputs, conv_kernel_3, scope='conv2')
            early_feature[scope_name] = layer_2

        scope_name = 'block_3'
        x_input = slim.max_pool2d(layer_2)
        num_outputs = 256

```

```

with tf.variable_scope(scope_name, reuse=reuse):
    layer_1 = slim.conv2d(x_input, num_outputs, conv_kernel_3, scope='conv1')
    layer_2 = slim.conv2d(layer_1, num_outputs, conv_kernel_3, scope='conv2')
    early_feature[scope_name] = layer_2

scope_name = 'block_4'
x_input = slim.max_pool2d(layer_2)
num_outputs = 512
with tf.variable_scope(scope_name, reuse=reuse):
    layer_1 = slim.conv2d(x_input, num_outputs, conv_kernel_3, scope='conv1')
    layer_2 = slim.conv2d(layer_1, num_outputs, conv_kernel_3, scope='conv2')
    early_feature[scope_name] = layer_2

scope_name = 'block_5'
x_input = slim.max_pool2d(layer_2)
num_outputs = 1024
with tf.variable_scope(scope_name, reuse=reuse):
    layer_1 = slim.conv2d(x_input, num_outputs, conv_kernel_3, scope='conv1')
    layer_2 = slim.conv2d(layer_1, num_outputs, conv_kernel_3, scope='conv2')
    early_feature[scope_name] = layer_2

scope_name = 'block_6'
num_outputs = 512
with tf.variable_scope(scope_name, reuse=reuse):
    trans_layer = slim.conv2d_transpose(
        layer_2, num_outputs, pool_kernel, pool_stride, scope='conv_trans')
    x_input = tf.concat([early_feature['block_4'], trans_layer], axis=3)
    layer_1 = slim.conv2d(x_input, num_outputs, conv_kernel_3, scope='conv1')
    layer_2 = slim.conv2d(layer_1, num_outputs, conv_kernel_3, scope='conv2')
    early_feature[scope_name] = layer_2

scope_name = 'block_7'
num_outputs = 256
with tf.variable_scope(scope_name, reuse=reuse):
    trans_layer = slim.conv2d_transpose(
        layer_2, num_outputs, pool_kernel, pool_stride, scope='conv_trans')
    x_input = tf.concat([early_feature['block_3'], trans_layer], axis=3)
    layer_1 = slim.conv2d(x_input, num_outputs, conv_kernel_3, scope='conv1')
    layer_2 = slim.conv2d(layer_1, num_outputs, conv_kernel_3, scope='conv2')
    early_feature[scope_name] = layer_2

scope_name = 'block_8'

```



```

num_outputs = 128
with tf.variable_scope(scope_name, reuse=reuse):
    trans_layer = slim.conv2d_transpose(
        layer_2, num_outputs, pool_kernel, pool_stride, scope='conv_trans')
    x_input = tf.concat([early_feature['block_2'], trans_layer], axis=3)
    layer_1 = slim.conv2d(x_input, num_outputs, conv_kernel_3, scope='conv1')
    layer_2 = slim.conv2d(layer_1, num_outputs, conv_kernel_3, scope='conv2')
    early_feature[scope_name] = layer_2

scope_name = 'block_9'
num_outputs = 64
with tf.variable_scope(scope_name, reuse=reuse):
    trans_layer = slim.conv2d_transpose(
        layer_2, num_outputs, pool_kernel, pool_stride, scope='conv_trans')
    x_input = tf.concat([early_feature['block_1'], trans_layer], axis=3)
    layer_1 = slim.conv2d(x_input, num_outputs, conv_kernel_3, scope='conv1')
    layer_2 = slim.conv2d(layer_1, num_outputs, conv_kernel_3, scope='conv2')
    early_feature[scope_name] = layer_2

scope_name = 'pred'
with tf.variable_scope(scope_name, reuse=reuse):
    layer_1 = slim.conv2d(layer_2, 1, conv_kernel_1, scope='conv1',
                          activation_fn=None, normalizer_fn=None)

    early_feature[scope_name] = layer_1
    pred = tf.sigmoid(layer_1)
return tf.squeeze(layer_1), tf.squeeze(pred)

def build_loss(x_in, y_in, H, phase):
    ...

    This function builds the loss and accuracy
    ...

    im_width = H['im_width']
    im_height = H['im_height']
    batch_size = H['batch_size']
    start_ind = H['start_ind']
    valid_size = H['valid_size']
    num_class = H['num_class']
    epsilon = H['epsilon']
    apply_class_balancing = H['apply_class_balancing']

```

```

logits, pred = build_pred(x_in, H, phase)
y_crop = tf.cast(tf.slice(y_in, begin=[0, start_ind, start_ind],
                           size=[-1, valid_size, valid_size]), tf.float32)
logits_crop = tf.slice(logits,
                       begin=[0, start_ind, start_ind],
                       size=[-1, valid_size, valid_size])
pred_crop = tf.cast(tf.slice(pred,
                              begin=[0, start_ind, start_ind],
                              size=[-1, valid_size, valid_size]), tf.float32)

if apply_class_balancing:
    class_weight = data_utils.calculate_class_weights()\
        [data_utils.CLASSES[class_type + 1]]
    intersection = tf.reduce_sum(tf.multiply(y_crop, pred_crop))
    union = tf.reduce_sum(y_crop) + tf.reduce_sum(pred_crop) - intersection
    loss = 1. - intersection / (union + tf.constant(epsilon))

pred_thres = tf.cast(tf.greater(pred_crop, 0.5), tf.float32)
inter = tf.reduce_sum(tf.multiply(tf.cast(y_crop, tf.float32), pred_thres))
uni = tf.reduce_sum(tf.cast(y_crop, tf.float32)) + \
    tf.reduce_sum(pred_thres) - inter
jaccard = inter / (uni + tf.constant(epsilon))

return loss, jaccard, logits_crop, pred_crop

```

```

def build(queues, H):
    ...

    This function returns the train operation, summary, global step
    ...

    im_width = H['im_width']
    im_height = H['im_height']
    num_class = H['num_class']
    num_channel = H['num_channel']
    batch_size = H['batch_size']
    log_dir = H['log_dir']
    norm_threshold = H['norm_threshold']

    loss, accuracy, x_in, y_in, logits, pred = {}, {}, {}, {}, {}, {}
    for phase in ['train', 'validate']:

```

```

x_in[phase], y_in[phase] = queues[phase].dequeue_many(batch_size)
loss[phase], accuracy[phase], logits[phase], pred[phase] = \
    build_loss(x_in[phase], y_in[phase], H, phase)

learning_rate = tf.placeholder(dtype=tf.float32)
opt = tf.train.AdamOptimizer(learning_rate=learning_rate, beta1=0.8, beta2=0.99)
global_step = tf.Variable(0, trainable=False)
tvars = tf.trainable_variables()
grads = tf.gradients(loss['train'], tvars)
grads, norm = tf.clip_by_global_norm(grads, norm_threshold)
update_ops = tf.get_collection(tf.GraphKeys.UPDATE_OPS)
with tf.control_dependencies(update_ops):
    train_op = opt.apply_gradients(zip(grads, tvars), global_step=global_step)

for phase in ['train', 'validate']:
    tf.summary.scalar(name=phase + '/loss', tensor=loss[phase])
    tf.summary.scalar(name=phase + '/accuracy', tensor=accuracy[phase])

    mean, var = tf.nn.moments(logits[phase], axes=[0, 1, 2])
    tf.summary.scalar(name=phase + '/logits/mean', tensor=mean)
    tf.summary.scalar(name=phase + '/logits/variance', tensor=var)

    mean, var = tf.nn.moments(pred[phase], axes=[0, 1, 2])
    tf.summary.scalar(name=phase + '/pred/mean', tensor=mean)
    tf.summary.scalar(name=phase + '/pred/variance', tensor=var)

summary_op = tf.summary.merge_all()
return loss, accuracy, train_op, summary_op, learning_rate, global_step

if __name__ == '__main__':
    hypes = './hypes/hypes.json'
    with open(hypes, 'r') as f:
        H = simplejson.load(f)
        im_width = H['im_width']
        im_height = H['im_height']
        num_class = H['num_class']
        num_channel = H['num_channel']
        queue_size = H['queue_size']
        save_iter = H['save_iter']
        print_iter = H['print_iter']
        class_type = H['class_type']

```

```

train_iter = H['train_iter']
lr = H['lr']
lr_decay_iter = H['lr_decay_iter']
log_dir = H['log_dir']
batch_size = H['batch_size']
now = datetime.datetime.now()

now_path = str(now.month) + '-' + str(now.day) + '_' + \
           str(now.hour) + '-' + str(now.minute) + '_' + H['loss_function']

sys.stdout.write('checkpoint name :{}'.format(now_path))
sys.stdout.write('\n')
sys.stdout.flush()

ckpt_path = os.path.join(log_dir, now_path, 'ckpt', 'ckpt')
hypes_path = os.path.join(log_dir, now_path, 'hypes')
summary_path = os.path.join(log_dir, now_path, 'summary')

for path in [ckpt_path, hypes_path, summary_path]:
    if not os.path.exists(path):
        os.makedirs(path)

def enqueue_thread(sess, data_gen, coord, phase, enqueue_op):
    while not coord.should_stop():
        img, label = data_gen.next()
        sess.run(enqueue_op, feed_dict={x_in[phase]: img, y_in[phase]: label})

x_in, y_in, queues, enqueue_op = {}, {}, {}, {}
shape = ((im_width, im_height, num_channel),
         (im_width, im_height))
for phase in ['train', 'validate']:
    x_in[phase] = tf.placeholder(dtype=tf.float32)
    y_in[phase] = tf.placeholder(dtype=tf.float32)
    queues[phase] = tf.FIFOQueue(
        capacity=queue_size, shapes=shape, dtypes=(tf.float32, tf.float32))
    enqueue_op[phase] = queues[phase].enqueue_many([x_in[phase], y_in[phase]])

loss, accuracy, train_op, summary_op, learning_rate, global_step = \
    build(queues, H)
data_gen = {}

```

```

for phase in ['train', 'validate']:
    is_train = {'train': True, 'validate': False}[phase]
    data_gen[phase] = train_utils.input_data(
        crop_per_img=1, class_id=class_type, reflection=True,
        rotation=360, train=is_train, crop_size=im_width)
    # Run the generator once to make sure the data is loaded into the memory
    # This will take a few minutes
    data_gen[phase].next()

sys.stdout.write('{} training images: {}\n'.format(
    len(train_utils.train_names), train_utils.train_names))
sys.stdout.write('\n')
sys.stdout.write('Training parameters: {}\n'.format(H))
sys.stdout.write('\n')
sys.stdout.flush()

with open(os.path.join(hypes_path, 'hypes.json'), 'w') as f:
    simplejson.dump(H, f)

config = tf.ConfigProto()
config.gpu_options.allow_growth = True
coord = tf.train.Coordinator()
threads = {}
saver = tf.train.Saver(max_to_keep=train_iter / save_iter + 1)

with tf.Session(config=config).as_default() as sess:
    summary_writer = tf.summary.FileWriter(logdir=summary_path, flush_secs=10)
    summary_writer.add_graph(sess.graph)
    for phase in ['train', 'validate']:
        threads[phase] = threading.Thread(
            target=enqueue_thread,
            args=(sess, data_gen[phase], coord, phase, enqueue_op[phase]))
        threads[phase].start()

    sess.run(tf.global_variables_initializer())
    start = time.time()
    for step in xrange(train_iter):
        if step and step % lr_decay_iter == 0:
            lr *= 0.1

        if step % print_iter == 0 or step == (train_iter - 1):

```

```

dt = (time.time() - start) / batch_size / print_iter
start = time.time()
_, train_loss, \
train_accuracy, validate_loss, \
validate_accuracy, summaries = \
    sess.run([train_op, loss['train'],
              accuracy['train'], loss['validate'],
              accuracy['validate'], summary_op],
              feed_dict={learning_rate: lr})
summary_writer.add_summary(
    summaries, global_step=global_step.eval())
str6 = 'Class ({}); '.format(class_type)
str0 = 'Global step ({}): LR: {1:0.5f}; '.format(
    global_step.eval(), lr)
str1 = 'Train loss {0:.2f}; '.format(train_loss)
str2 = 'Train accuracy {}%; '.format(int(100 * train_accuracy))
str3 = 'Validate loss {0:.2f}; '.format(validate_loss)
str4 = 'Validate accuracy {}%; '.format(
    int(100 * validate_accuracy))
str5 = 'Time / image: {0:0.1f}ms'.format(1000 * dt if step else 0)
sys.stdout.write(
    str6 + str0 + str1 + str2 + str3 + str4 + str5 + '\n')
sys.stdout.flush()
else:
    sess.run([train_op, loss['train']], feed_dict={learning_rate: lr})

if step % save_iter == 0 or step == (train_iter - 1):
    saver.save(sess, ckpt_path, global_step=global_step.eval())

```